PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

Horácio Aaron Christhian Galdezanni Pedroso

Classificação de Sentenças em Chats de Dispositivos Móveis em Análises Periciais

MESTRADO EM
CIÊNCIA DA COMPUTAÇÃO
PUCPR

CURITIBA
2025

Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática

Classificação de Sentenças em Chats de Dispositivos Móveis em Análises Periciais

Horácio Aaron Christhian Galdezanni Pedroso

Dissertação apresentada ao Programa de Pós-Graduação em Informática como requisito parcial para obtenção do título de Mestre em Informática.

Campo de Concentração: Ciência da Computação

Orientador: Jean Paul Barddal

Curitiba
2025

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR -- Biblioteca Central

P372c
2025

Pedroso, Horácio Aaron Christhian Galdezanni
Classificação de sentenças em chats de dispositivos móveis em análises periciais / Horácio Aaron Christhian Galdezanni Pedroso ; orientador: Jean Paul Barddal. 2025
141 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2025
Bibliografia: f.133-141

1. Informática. 2. Ciência forense digital. 3. Algoritmos computacionais. 4. Perícia (Exame técnico). 5. Laudos periciais. I. Barddal, Jean Paul. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título

CDD 20. ed. – 004

Bibliotecária: Edilene de Oliveira dos Santos CRB 9 / 1636

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 13 de fevereiro de 2026.

07-2026

DECLARAÇÃO

Declaro para os devidos fins, que **HORÁCIO AARON CHRISTHIAN GALDEZANNI PEDROSO** defendeu a dissertação de Mestrado intitulada **"Classificação de Sentenças em Chats de Dispositivos Móveis em Análises Periciais"**, na área de concentração Ciência da Computação no dia 01 de dezembro de 2025, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

_____
Prof. Dr. Jean Paul Barddal
Coordenador do Programa de Pós-Graduação em Informática

Dedico este trabalho à minha mãe Maria Aparecida (in memoriam), quem entregou tudo de si para que eu pudesse caminhar...

"O Senhor com sabedoria fundou a terra, com inteligência estabeleceu os céus. Pelo seu conhecimento os abismos se rompem, e as nuvens destilam orvalho."

Provérbios 3:19-20

Resumo

O crescimento e facilidade de acesso a novas tecnologias de comunicação trouxe muitos benefícios à sociedade, todavia acabou gerando externalidades negativas, como o uso dessas inovações em atividades criminosas. Esse cenário acaba repercutindo na atividade pericial, o aumento do volume de equipamentos e dados digitais analisados em perícias forenses aumenta a carga de trabalho das equipes especializadas. Nesse contexto, o uso de técnicas de Inteligência Artificial surge como alternativa para apoiar a análise de conversas em celulares e a detecção de indícios de crime. Esta pesquisa aplica métodos de *Machine Learning* (ML) e Processamento de Linguagem Natural (NLP) para avaliar a capacidade de identificar envolvimento criminoso em mensagens de texto e áudio no WhatsApp, por meio de algoritmos de classificação de sentenças. O desenvolvimento da pesquisa envolveu a organização dos dados e a geração de uma base rotulada, construída a partir de nove casos reais disponibilizados por peritos da Polícia Civil do Estado do Paraná (PCPR), que totalizaram 392.416 sentenças e 2.961.033 palavras obtidas a partir da troca de mensagens desses aparelhos. A partir disso, foi possível a avaliação de Grandes Modelos de Linguagem (LLM) para a tarefa de classificação de sentenças, além da aplicação de técnicas de *fine-tuning* em modelos pré-treinados e treinamento de classificadores, bem como a condução de experimentos em trinta diferentes cenários, combinando variações no tipo de balanceamento, forma de representação e variável de entrada. Como resultado, o melhor desempenho foi obtido pelos modelos supervisionados, em especial quando se aplicou a sobreamostragem, utilizando a representação *Term-Frequency Inverse Document Frequency* (TF-IDF) utilizando a concatenação de keyphrase e sentence como variável de treinamento, combinada ao classificador XGBoost. Essa configuração alcançou um *F1-Score* médio de 0,663 nos nove casos analisados, representando um aumento de 20,11% em relação à técnica atualmente empregada pela instituição, baseada em busca de palavras-chave. Tais resultados evidenciam o potencial do uso de modelos de linguagem natural para apoiar de forma mais eficaz a atividade pericial criminal.

Palavras-chave: Classificação de Sentenças, Perícia Forense

Abstract

The growth and accessibility of new communication technologies have brought many benefits to society; however, they have also generated negative externalities, such as their use in criminal activities. This scenario has a direct impact on forensic practice, as the increasing volume of equipment and digital data examined in investigations has raised the workload of specialized teams. In this context, Artificial Intelligence techniques emerge as an alternative to support the analysis of mobile phone conversations and the detection of crime-related evidence. This research applies *Machine Learning* (ML) and Processamento de Linguagem Natural (NLP) methods to assess their ability to identify criminal involvement in text and audio messages exchanged via WhatsApp, using sentence classification algorithms. The development of this research involved organizing the data and generating a labeled, dataset built from nine real cases provided by forensic experts from Polícia Civil do Estado do Paraná (PCPR), these cases totaled 392,416 sentences and 2,961,033 words extracted from the message exchanges found on those devices. Based on this dataset, we evaluated Grandes Modelos de Linguagem (LLM) for the task of sentence classification, applied fine-tuning techniques on pre-trained models, and trained classifiers, conducting experiments across thirty different scenarios that combined variations in sampling strategies, feature representation, and input variables. As a result, the best performance was achieved by supervised models, especially when oversampling was applied, using *Term-Frequency Inverse Document Frequency* (TF-IDF) representation with concatenated keyphrases and sentences as input variables, combined with the XGBoost classifier. This configuration reached an average *F1-Score* of 0.663 across the nine analyzed cases, representing a 20.11% improvement compared to the keyword-based technique currently employed by the institution. These results highlight the potential of natural language models to more effectively support forensic criminal investigations.

Key-words: Sentence Classification, Forensic Expertise

Agradecimentos

A Deus, pela vida, por Sua graça e pelo sustento constante.
À minha mãe (*in memoriam*), por seu amor e por sempre estar ao meu lado, sendo em todos os momentos o lugar onde pude encontrar paz, força e sabedoria.
Ao professor Jean Paul Barddal, pelo conhecimento transmitido, orientação, dedicação e apoio que viabilizou a conclusão deste trabalho.
À professora Cínthia Obladen de Almendra Freitas, pela coordenação do projeto SICRET II.
Ao Dr. Jeovane Honório Alves, pela valiosa contribuição ao projeto, compartilhando conhecimento e experiência sobre o tema.
À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo fomento que tornou possível a execução do projeto SICRET II e o avanço desta pesquisa.
À Pontifícia Universidade Católica do Paraná (PUCPR), pela infraestrutura, pelo suporte institucional e pela oportunidade de realizar esta pesquisa em um ambiente acadêmico de excelência.
À Polícia Científica do Paraná - PCPR, especialmente ao perito Joel Eduardo Matschinske Köster, pelo apoio fundamental e pela orientação durante todo o período de concepção e experimentação da pesquisa.
Ao Programa de Pós-Graduação em Informática - PUCPR (PPGIA), pela jornada de aprendizado, pelo rigor científico e pelo conhecimento compartilhado ao longo do curso
A todos que, de alguma forma, colaboraram e contribuíram para a realização deste estudo.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ADB Android Debug Bridge

ARFIMA Autoregressive Fractionally Integrated Moving Average

ANATEL Agência Nacional de Telecomunicações

BERT Bidirectional Encoder Representations for Transformers

BTM Biterm Topic Model

CoT chain-of-thought

CSV Comma Separated Values

CAPES Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

DAPT *Domain-adaptive pre-training*

DDoS Distributed Denial of Service

DNN *Deep Neural Networks*

GPT Generative Pre-Training

GPU Graphics Processing Unit

HAR Heterogeneous Autoregressive Model

HMM Hidden Markov Models

HO Hold Out

HPO *Hyperparameter Optimization*

ICL *in-context learning*

KPE *Key Phrase Extraction*

LLM Grandes Modelos de Linguagem

LSTM *Long Short-Term Memory*

LOGO *Leave One Group Out*

ML *Machine Learning*

MLM *Masked Language Model*

MLP *Multilayer Perceptron*

NER *Named Entity Recognition*

NLM Modelos de Linguagem Neural

NLP Processamento de Linguagem Natural

NSP *Next Sentence Prediction*

NUCIBER Núcleo de Combate aos Cibercrime

PCPR Polícia Civil do Estado do Paraná

PCP/PR Polícia Científica do Estado do Paraná

PLM *Pre-trained Language Model*

PPGIA Programa de Pós-Graduação em Informática - PUCPR

PUCPR Pontifícia Universidade Católica do Paraná

RAKE *Rapid Automated Keyword Extraction*

REGEX *Regular Expressions*

RLHF *reinforcement learning from human feedback*

RNN *Recurrent Neural Networks*

ROS *Random Oversampling*

RUS *Random Undersampling*

RTE *Recognizing Textual Entailment*

SiCRet Sistema de Cruzamento de Registros Telefônicos

SLM Modelos Estatísticos de Linguagem

STS *Sentence Textual Similarity*

SVM *Support Vector Machine*

TF-IDF *Term-Frequency Inverse Document Frequency*

XGBoost *Extreme Gradient Boosting*

1

INTRODUÇÃO

O desenvolvimento tecnológico, especialmente nas últimas décadas, proporcionou grandes mudanças no estilo de vida da sociedade, incluindo em seus meios de comunicação.

O avanço das telecomunicações encurtou espaços e acelerou a troca de informações dado que o que só era possível ser feito de forma analógica passou a ser realizado digitalmente. Esse processo foi ainda impulsionado com a democratização da internet, bem como da possibilidade de acesso desta por meio de *smartphones* permitindo que tal interação fosse possível a qualquer momento e lugar.

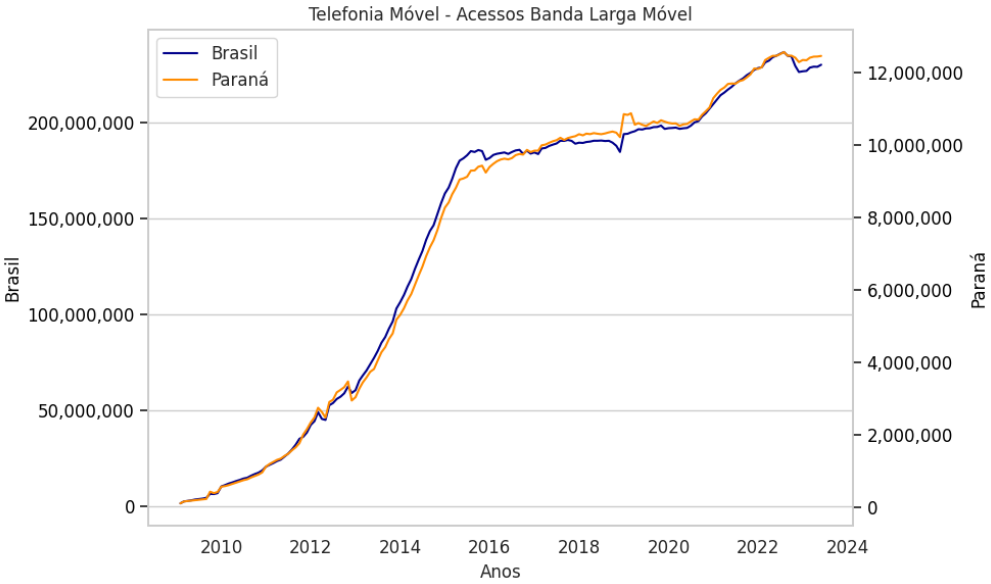
Essa inovação possibilitou o surgimento de novas modalidades de comunicação, inclusive em massa, com a introdução de redes sociais, as quais tiraram a exclusividade de uma difusão (*broadcast*) de apenas algumas entidades – como emissoras de rádio e televisão – e a popularizaram permitindo que qualquer indivíduo a faça a partir de um aparelho celular com conexão à internet.

Dados da Agência Nacional de Telecomunicações (ANATEL), conforme Figura 1.1, traduzem essa evolução de forma que seja possível perceber esse avanço a partir do ano de 2009, com destaque ao *boom* ocorrido entre os anos de 2013 e 2015, com o aumento de quase 300% e cerca de 305% nos acessos de banda larga móvel no Brasil e no Paraná, respectivamente.

Neste intervalo pode-se perceber o significativo incremento nos acessos, todavia esse aumento permanece – ainda que de maneira menos acelerada – até os dias atuais.

Concomitantemente à facilidade de acesso aos meios de comunicação, novos

Figura 1.1: Telefonia Móvel - Acessos Banda Larga Móvel



Fonte: Elaborado pelo autor, com base em ANATEL (2023).

desafios são apresentados. Entre eles, uma questão latente é que estas tecnologias também acabaram se tornando meios de comunicação facilitados e amplamente utilizados por organizações criminosas ou até mesmo por indivíduos que atuam isoladamente em ações delituosas. Assim, com a mesma agilidade que o desenvolvimento na comunicação fora proporcionada à sociedade, o mundo do crime se apropriou e se especializou neste tema com o intuito de dificultar sua detecção e potencializar seu alcance.

Todavia, quando se fala sobre atividades criminosas utilizando tecnologia – cibercrimes – tem-se diversos níveis e formas de abuso, ao que McQuade (2009) categoriza em:

1. Usuários Negligentes: Violam políticas de segurança, ou não praticam segurança da informação;

2. Criminosos Tradicionais: Utilizam computadores para apoiar atividades ilícitas convencionais;

3. Fraudadores: Aplicam técnicas de *phishing*, *spoofing* ou *spam* enriquecimento ilícito;

4. *Hackers*, Invasores de Senhas e Criadores de Pragas: Usam computadores para expor ou explorar vulnerabilidades ilegalmente;
5. Autores e Distribuidores de Códigos Maliciosos: Criam copiam ou espalham programas nocivos;
6. Piratas de Música, Filmes e Programas: Infringem direitos autorais através do uso de tecnologia;
7. Atacantes de Sistemas: Realizam ataques de negação de serviço com intuito de incapacitar, ou danificar sistemas;
8. Perseguidores, Pedófilos e Outros Ofensores Cibernéticos: Cometem ofensas sexuais ou assédio online;
9. Trapaceiros Acadêmicos: trapaceiam em exames, ou fraudam pesquisas com emprego de tecnologia da informação;
10. Criminosos Organizados: Utilizam de tecnologia em atividades criminosas organizadas;
11. Espiões Corporativos, Governamentais e Autônomos: utilizam ferramentas para espionar governos, empresas ou fins pessoais;
12. Ciberterroristas: Buscam instigar medo ou danificar infraestrutura crítica.

Importa ressaltar que essa lista de cibercrimes não possui, necessariamente, uma relação hierárquica entre elas. Independentemente de tal categorização, equipamentos informáticos podem ser utilizados na realização de crimes. Ainda segundo McQuade (2009), esses podem ser divididos em três categorias de acordo com o modo no qual tal dispositivo fora empregado:

1. O equipamento desempenha o papel de uma arma de crime, como no caso de Distributed Denial of Service (DDoS) (ataques de negação de serviço);
2. O dispositivo é alvo de um crime, como por exemplo, na invasão de um computador;
3. O aparelho é usado como facilitador de um crime, como no uso de um computador para armazenar dados incriminadores.

Tendo em vista as categorias de crimes listadas anteriormente, fica evidente que o envolvimento criminoso com as novas tecnologias se tornou tão difundido e especializado que os próprios departamentos policiais passaram a criar áreas voltadas especificamente na investigação e análise pericial de dessa ordem.

No Estado do Paraná, por exemplo, foi criado em 2005, o Núcleo de Combate aos Cibercrime (NUCIBER), conforme Assembleia Legislativa do Estado do Paraná (2023), órgão vinculado à PCPR, especializado na detecção e combate a crimes cibernéticos.

Além disso a própria Polícia Científica do Estado do Paraná (PCP/PR) dispõe de equipe pericial especializada na análise forense de dados eletrônicos obtidos a partir de dispositivos sob análise ou investigação policial, inclusive contando com investimento Polícia Científica do Estado do Paraná (2023a) em equipamentos dedicados para tal fim.

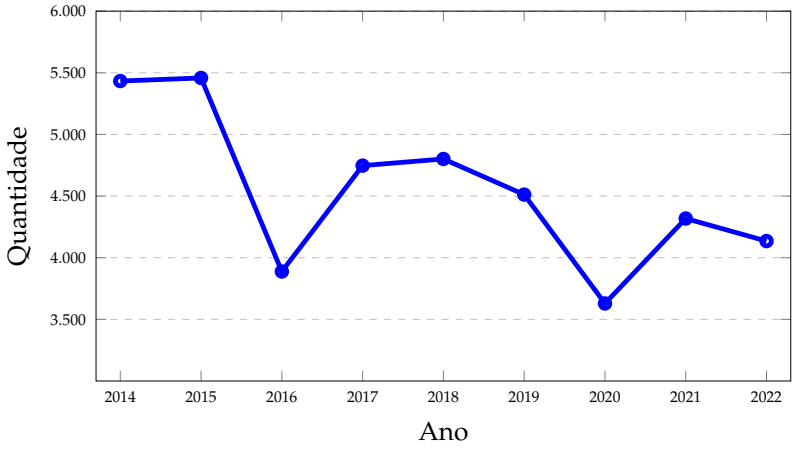
Para Hassan (2019), a perícia digital pode ser agrupada de acordo com o tipo de evidência obtida, como: computadores, *laptops*, dispositivos de armazenamento digital, sistemas operacionais, aplicativos, *logs* de acesso, bem como celulares, *tablets*, além de fluxo de dados de rede, banco de dados, computação em nuvem, entre outros.

Dada essa gama de possibilidades e possíveis ramificações ligadas aos cibercrimes, o poder público vem investido com vistas a fazer frente às novas formas de atividades criminosas praticadas, ao ponto de as unidades polícias dedicarem esforços especializados a esse *modus operandi*, dada sua diversificação e massa de conteúdo submetido à averiguação, especialmente com relação a dispositivos móveis.

Corrobora com essa colocação o volume de perícias digitais realizadas pela polícia. Conforme informações da PCP/PR entre os anos de 2014 e 2022, em geral houve uma média anual de 4.500 peças examinadas, como apresentado na Figura 1.2.

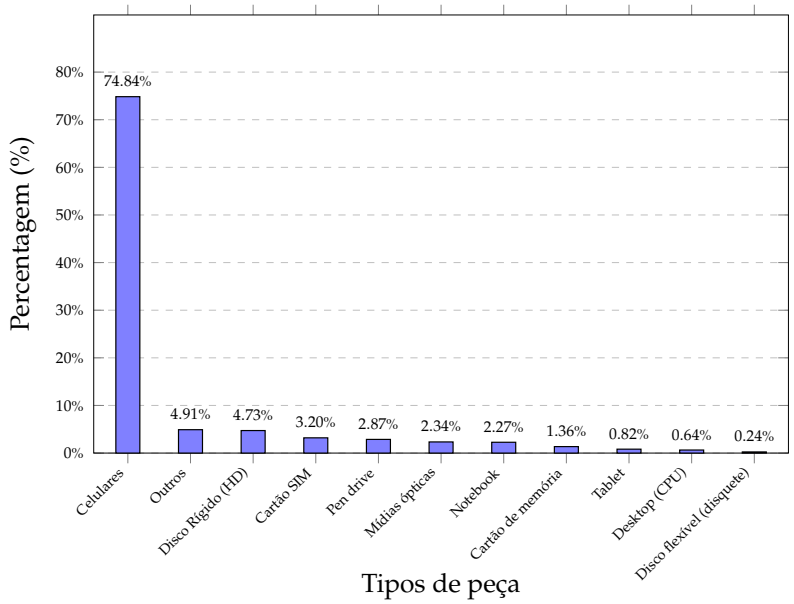
Todavia, dentre esses tipos de peças os dispositivos móveis representaram quase 75% do total de equipamentos analisados, um dado expressivo que se destaca na Figura 1.3, tendo inclusive atingido um número recorde de peças examinadas, 584 unidades em janeiro de 2024 (Agência Estadual de Notícias, 2024).

Figura 1.2: Quantidade de peças (celulares) - PCP/PR



Fonte: Elaborado pelo autor, com base em Polícia Científica do Estado do Paraná (2023b).

Figura 1.3: Tipo de peça analisada (anos 2014-2022) - PCP/PR

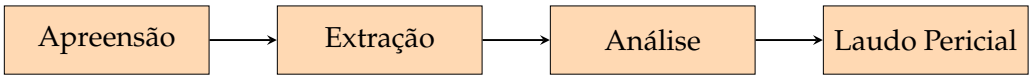


Fonte: Elaborado pelo autor, com base em Polícia Científica do Estado do Paraná (2023b).

Esses materiais, uma vez apreendidos, são submetidos a um processo de análise forense digital, elemento crucial para a conclusão da perícia. Tal processo, conforme compreendido pelos autores em Hassan (2019), divide-se em

quatro etapas principais. A Figura 1.4 ilustra o fluxo desse processo, estrutura essa fundamental para a sistematização da análise, assegurando que todas as evidências digitais sejam examinadas de maneira abrangente e conforme os padrões estabelecidos pela ciência forense.

Figura 1.4: Etapas do processo de forense digital



Fonte: Elaborado pelo autor, com base em Hassan (2019).

Com relação ao explicitado na Figura 1.4, observa-se que a etapa de apreensão trata-se da obtenção do dispositivo em si, isto é, da evidência física. Na sequência, é efetuada a extração, ou duplicação dos dados do dispositivo. Durante a fase de análise, o perito tem seu foco em procurar pistas de indícios de envolvimento criminoso.

Por fim, o laudo pericial é quando o perito expõe, de forma estruturada, todas suas descobertas, evidências, métodos e ferramentas empregadas.

Ressalta-se, que a presente pesquisa tem seu enfoque no auxílio do processo pericial, especialmente no que tange a atividade de **análise**, visando identificar possíveis indícios que possam sinalizar envolvimento em atividades ilícitas, por meio da aplicação de técnicas de *Machine Learning* (ML), mais especificamente a classificação de sentenças.

Aprofundando o contexto supra mencionado, o crescente volume de aparelhos envolvidos e a quantidade de dados contida em cada dispositivo, bem como a complexa natureza dos dados extraídos de celulares apreendidos acabam se tornando um desafio para análise pericial, dada a atual multimodalidade das comunicações englobando textos, áudios, imagens e vídeos.

Além disso, há ainda a necessidade de identificar, dentro de uma gama de distintas conversas, quais são relacionadas a alguma atividade criminosa, para que esta possa servir de evidência ou insumo a fim de colaborar na investigação policial.

Assim, com objetivo de identificar trocas de mensagens que possam indicar possível envolvimento criminoso, emerge o interesse de empregar técnicas computacionais avançadas para análise textual, como no caso de detecção através de busca de palavras-chave ou pela aplicação de filtros específicos.

Esse cenário tem mobilizado a comunidade científica a fim de explorar mo-

delos de NLP com objetivo de auxiliar nas atividades de perícia forense no combate à atividades ilegais (RODRIGUES et al., 2022).

A necessidade de implementar modelos de Processamento de Linguagem Natural (NLP) surge da característica da linguagem natural ser intrinsecamente acessível aos seres humanos, mas não às máquinas, havendo a necessidade de que haja uma modelagem capaz de converter um texto a algo inteligível à ela (ZHAO et al., 2023).

Destaca-se que a NLP visa superar essa lacuna, facilitando a interação lexical entre humanos e máquinas. Esse campo procura desenvolver algoritmos e modelos capazes de interpretar, compreender e gerar linguagem natural, possibilitando assim uma comunicação efetiva entre usuários e sistemas computacionais.

Ao longo dos anos, a área de Processamento de Linguagem Natural vem se desenvolvendo introduzindo novos modelos a partir de diferentes paradigmas de modelagem, segundo Zhao et al. (2023), como os modelos estatísticos, redes neurais, de bases pré-treinadas, até os LLM - *Large Language Models*, tendo cada um com sua aplicação e contribuição para desenvolvimento deste campo da computação.

A partir desse panorama apresentado, e tendo em vista a necessidade de buscar novas formas que permitam a automatização e o auxílio em atividades da análise pericial digital, o presente trabalho visa a aplicação e avaliação de desempenho de modelos e técnicas de NLP na classificação de sentenças de mensagens de texto obtidas de celulares apreendidos em investigações criminais, a fim de verificar a possibilidade de identificação de conversas passíveis de qualquer relacionamento com atividades criminosas.

Assim, a pesquisa se propõe a identificar um conjunto de modelo e técnica de NLP que permita auxiliar nas atividades de identificação de mensagens suspeitas, ligadas ao crime, considerando a especificidade situacional dos dados analisados, em um diálogo entre a ciência de dados e a ciência forense.

1.1 Formulação do Problema

Assim como abordado anteriormente, a principal motivação deste trabalho consiste em auxiliar a análise pericial da PCP/PR, proporcionando maior agilidade e abrangência em suas análises. Nesse contexto, as técnicas de Processamento de Linguagem Natural (NLP) aqui empregadas possuem a capacidade

de analisar as conversações obtidas nos equipamentos digitais em procedimento pericial, contribuindo para inferências sobre textos associados a conteúdos específicos — neste caso, conteúdos potencialmente criminosos, como por exemplo, tráfico de drogas.

Ressalta-se que o projeto não se restringe às trocas de mensagens textuais, uma vez que mensagens de áudio também integram o escopo de análise, representando um avanço no contexto de pesquisas forenses aplicadas a dados multimodais. Para tal, os áudios são convertidos em texto e incorporados ao fluxo conversacional, respeitando a ordem cronológica dos eventos, conforme detalhado no Capítulo 4. Essa estratégia permite ampliar a contextualização das interações e viabiliza o tratamento de detalhes relevantes sobre dinâmicas e padrões associados à atuação criminal.

Conforme argumentado por Hassan (2019), o processo de perícia forense pode ser estruturado em quatro atividades principais, a saber: apreensão, extração, análise e laudo pericial.

A partir da observação da execução da rotina pericial, nota-se na fase de análise um maior dispêndio temporal e laboral dada gama de mensagem (textos, áudios e imagens) que fazem parte do escopo de atividade do perito forense, ainda mais com o crescimento e diversificação nos modos de comunicação hodiernos.

Desta forma, em suas atribuições periciais, cabe ao agente identificar possíveis indícios de algum envolvimento em atividade criminosa. Neste ponto esta pesquisa visa auxiliar a atividade do perito, onde modelos de NLP, especificamente de classificação de sentenças, são empregados a fim de indicar qual mensagem possa sugerir tal envolvimento.

Ressalta-se que o projeto não se detém apenas nas trocas de mensagens textuais, como também os áudios são parte da análise. Para tal, esses são convertidos em textos e introduzidos às mensagens textuais respeitando a ordem cronológica, como detalhado no Capítulo 4.

Importa ressaltar o emprego destas técnicas dada a quantidade de textos analisados em cada perícia. Onde, apenas com intuito ilustrativo, o conteúdo dos 9 celulares periciados analisados pelo projeto – base de utilizada na pesquisa – de casos reais da Polícia Científica do Estado do Paraná (PCP/PR), continham em média 329.004 palavras por dispositivo, variando entre cerca de 1.196 e 1.301.023 palavas para cada análise pericial.

Com objetivo de alcançar um resultado mais assertivo nesta análise, este es-

tudo testa diversas técnicas e modelos de linguagem natural, inclusive partindo, a princípio, do uso de técnicas de extração de palavras-chave, frases-chave e finalmente a classificação de sentenças.

Convém destacar que a problemática computacional envolvida neste projeto se refere à possibilidade de ajuste de um modelo de NLP capaz de lidar com textos que se caracterizam pelo uso de linguagem informal, erros ortográficos e neologismos, uma vez que os modelos atualmente disponibilizados em português não possuem grande eficácia na análise de textos que possuem uma identidade própria, codificada, devido seu envolvimento criminoso.

Além da problemática léxica supramencionada, outra situação a ser tratada fora a organização dessas mensagens em ordem cronológica e identificando seus emissores e receptores, quer em trocas individuais ou coletivas (grupos de troca de mensagens), indicativos extremamente relevantes à análise pericial.

Para aprimorar a precisão dos modelos empregados, foi essencial desenvolver um *corpus* rotulado especificamente para esse projeto. Isso envolveu a classificação manual de sentenças com potencial indício de envolvimento em atividades criminosas, para que os algorítimos pudessem compreender o contexto ao qual estavam sendo aplicados.

A base de dados foi constituída por 9 casos reais (conteúdo de 9 aparelhos de celular apreendidos e periciados), originárias de dispositivos móveis apreendidos e examinados pela PCP/PR, todas vinculadas a investigações concentradas no combate ao tráfico de drogas.

1.2 Motivação

A motivação desta pesquisa emerge da necessidade de obtenção de um mecanismo que auxilie a atividade de perícia forense devido à grande quantidade de dados geralmente extraídos e analisados, que frequentemente apresentam características de linguagem informal e estruturada de forma inconsistente, como já discorrido.

A relevância aplicada do tema e a viabilidade desta investigação também se relacionam ao Programa de Cooperação Acadêmica em Segurança Pública e Ciências Forenses da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por meio do Edital nº 16/2020 (BRASIL, 2020).

Neste sentido, esta dissertação busca ajustar, testar e comparar diversos mo-

delos de NLP e de classificação de sentenças, avaliando qual combinação é capaz de gerar melhor desempenho aplicado ao contexto de troca de mensagens ligadas à atividades criminosas.

1.3 Objetivos

O objetivo deste trabalho é desenvolver um modelo para classificação de sentenças a partir de textos de trocas de mensagens no contexto de análises periciais forenses, para auxílio na identificação de mensagens extraídas de dispositivos móveis em procedimentos periciais que possam estar relacionadas com atividades criminosas.

Para exequibilidade desse objetivo, foram definidos os objetivos específicos abaixo:

1. Gerar base de dados rotulados sobre sentenças com possível envolvimento criminal;
2. Comparar diferentes modelos classificadores de distintos tipos de arquitetura;
3. Levantar uma base específica para treinamento na aplicação da técnica de *fine-tuning*;
4. Elaborar um protocolo de classificação de sentenças com indícios de envolvimento criminoso;
5. Avaliar a capacidade de classificação de sentenças de modelos Grandes Modelos de Linguagem (LLM) para os casos reais da pesquisa;
6. Verificar a eficácia do ajuste de modelo (*fine-tuning*) a partir de uma base de dados extraídos de mensagens reais.

1.4 Questões de Pesquisa

A partir do tema desta pesquisa relacionada à extração de frases-chave, foram identificadas as seguintes questões:

1. As técnicas de classificação de sentenças baseadas em modelos de aprendizado de máquina são mais eficazes do que a busca por palavras-chave na identificação de indícios de atividades criminosas em mensagens textuais?

2. Qual a eficácia dos modelos LLM na execução da rotulação de base de dados relativo à sentenças com conteúdos que possam indicar alguma atividade criminosa?

3. Ajuste fino (*fine-tuning*) em modelos de NLP pré-treinados potencializa a extração de conhecimento de textos relacionados à perícia forense?

1.5 Apoio institucional e restrições de confidencialidade

A condução desta pesquisa foi possível a partir de uma articulação institucional com a Polícia Científica do Estado do Paraná (PCP/PR), que viabilizou o acesso a dados reais provenientes de exames periciais — condição essencial para a experimentação e o treinamento dos modelos propostos. Além de permitir o uso de uma base representativa do contexto forense, essa cooperação incluiu o acompanhamento técnico de peritos, que contribuíram na seleção e categorização dos materiais, bem como na análise crítica dos resultados obtidos, aproximando as decisões metodológicas das demandas práticas da perícia digital.

Essa colaboração se insere em uma trajetória iniciada no projeto Sistema de Cruzamento de Registros Telefônicos (SiCRet) Grochocki et al. (2013), concebido com o objetivo de desenvolver um sistema capaz de, a partir de uma base pericial estruturada, realizar o cruzamento de informações de diferentes dispositivos examinados. A proposta buscava identificar propriedades, padrões e comportamentos dos dados extraídos, de modo a evidenciar relações e dinâmicas relevantes para a compreensão do fluxo criminoso no ambiente digital. No âmbito dessa continuidade, a presente dissertação avança no uso de técnicas de ciência de dados e NLP para apoiar a triagem e a análise de grandes volumes de mensagens, com foco na identificação automatizada de conteúdo potencialmente ilícito.

Em sua etapa atual, o esforço de pesquisa integra o **SICRET II** (forense digital, ciência de dados e inteligência artificial aplicadas aos laudos periciais

de dispositivos móveis) e conta com apoio da CAPES por meio do Programa de Cooperação Acadêmica em Segurança Pública e Ciências Forenses (BRASIL, 2020). Esse apoio reforça a relevância do tema e contribui para a consolidação de uma agenda de pesquisa aplicada, orientada à modernização de métodos periciais e à redução do custo operacional associado à análise manual de dados digitais.

Vale destacar, que pelo fato de estar tratando com dados reais, relacionados a processos de investigação policial, há uma imposição de carácter sensível e confidencial, inserindo requisitos especiais a serem respeitados como processamento *on-premise* e não transferência ou exposição de dados e informações para fora das infraestruturas físicas e lógicas da PCP/PR.

Importa destacar que o trabalho com dados reais vinculados a procedimentos de investigação impõe requisitos rigorosos de confidencialidade e segurança da informação. Por essa razão, foram observadas restrições específicas, como a necessidade de processamento *on-premise* e a vedação de transferência, exposição ou compartilhamento de dados e metadados fora das infraestruturas físicas e lógicas da Polícia Científica do Estado do Paraná (PCP/PR). Tais condicionantes influenciam diretamente escolhas metodológicas e tecnológicas do estudo, exigindo soluções que conciliem desempenho, reprodutibilidade e conformidade com os protocolos institucionais de proteção de dados sensíveis.

Por fim, cabe destacar que a natureza sigilosa do material analisado implicou requisitos adicionais ao desenvolvimento desta pesquisa. Primeiramente, todas as atividades foram realizadas exclusivamente nas instalações da PCPR e utilizando seus equipamentos, com acompanhamento presencial de um perito durante as sessões de trabalho.

Adicionalmente, adotou-se uma estratégia de anonimização compatível com o escopo do estudo: como a análise concentrou-se exclusivamente no conteúdo textual das mensagens, não foram utilizados, em nenhum momento, dados pessoais ou identificadores dos indivíduos relacionados aos dispositivos periciados. Assim, o enfoque permaneceu restrito às mensagens trocadas, preservando-se a confidencialidade dos envolvidos e mitigando riscos de exposição de informações sensíveis.

1.6 Estrutura do Documento

O presente trabalho está estruturado da seguinte forma:

- O Capítulo 1 apresenta a contextualização do problema, sua relevância no contexto pericial, a motivação para o estudo, além dos objetivos e questões de pesquisa que nortearam a investigação;
- O Capítulo 2 contempla a revisão de literatura, aprofundando os fundamentos teóricos de NLP e dos algoritmos de classificação utilizados
- O Capítulo 3 revisa os principais trabalhos relacionados à aplicação dessas técnicas em contextos forenses e investigativos;
- O Capítulo 4 descreve a metodologia desenvolvida, detalhando todo processo empregado na pesquisa com as etapas de extração, transcrição, rotulação, balanceamento, representação, treinamento e validação dos modelos;
- O Capítulo 5 traz o protocolo experimental adotado, detalhando os cenários testados, os dados utilizados e os critérios de avaliação empregados;
- O Capítulo 6 apresenta e discute os resultados obtidos a partir da validação por grupos, comparando o desempenho dos modelos, exibe e analisa o modelo selecionado, destacando os pontos mais relevantes possam ter contribuído para que obtivesse tal desempenho;
- Por fim, o Capítulo 7 traz as considerações finais, limitações do estudo e sugestões para trabalhos futuros.

1.7 Considerações Finais

Este capítulo apresentou a contextualização do problema de pesquisa, destacando o crescimento das tecnologias de comunicação e sua apropriação por atividades criminosas, gerando um aumento significativo na carga de trabalho das equipes de perícia digital, especialmente na análise de dispositivos móveis.

Diante desse cenário, a pesquisa propôs a aplicação e avaliação de métodos de *Machine Learning* (ML) e Processamento de Linguagem Natural (NLP) para a

classificação de sentenças em mensagens de texto e áudio trocadas no aplicativo *WhatsApp*, buscando identificar indícios de envolvimento criminoso.

O trabalho envolveu a organização de dados e a geração de uma base rotulada a partir de nove casos reais fornecidos pelaPolícia Civil do Estado do Paraná (PCPR). Com o problema e os objetivos de pesquisa definidos, o estudo se propõe, no próximo capítulo, a apresentar o arcabouço teórico necessário, introduzindo os conceitos fundamentais de processamento de linguagem, seus modelos computacionais e os algoritmos de classificação que sustentam a solução proposta, conforme projeto de pesquisa aprovado no Edital nº 16/2020 PROCAD/SPCF da CAPES (BRASIL, 2020).

2

FUNDAMENTAÇÃO TEÓRICA

O presente capítulo tem por objetivo apresentar o arcabouço teórico e os trabalhos correlatos que sustentam esta pesquisa. Para tanto, são introduzidos os principais conceitos ligados à Processamento de Linguagem Natural (NLP), seus modelos computacionais e classificadores, bem como estudos anteriores que aplicaram essas técnicas em contextos relacionados à ciência forense, especialmente no âmbito da análise de mensagens extraídas de dispositivos móveis.

Primeiramente, são apresentados os fundamentos técnicos que embasam o uso de processamento de linguagem e suas aplicações em tarefas como a extração de palavras-chave e a classificação de sentenças, incluindo uma explanação sobre os modelos estatísticos e os modelos baseados em aprendizado de máquina (ML), destaque para os modelos de linguagem pré-treinados (*Pre-trained Language Model* (PLM) e Grandes Modelos de Linguagem (LLM)).

Na sequência são apresentados os trabalhos relacionados que ilustram a aplicação dessas técnicas no contexto pericial e criminal, com destaque para a identificação de sentenças com indícios de atividade criminosa.

A revisão aqui conduzida visa, portanto, apresentar a base conceitual que sustenta este trabalho, bem como o pano de fundo sob o qual se desenvolveu a análise textual ao longo dos anos, além de demonstrar as lacunas e oportunidades que justificam a realização da presente pesquisa.

Dada a natureza técnico-computacional da pesquisa, com foco na análise de mensagens aplicada à atividade pericial, importa formalizar os termos e estruturas envolvidas no NLP, estabelecendo uma base sólida para compreensão das etapas de pré-processamento, modelagem e classificação de sentenças.

São abordados, inicialmente, as conceitos fundamentais sobre a composição textual, como *tokens*, sentenças, sequências e *embeddings*, além da distinção entre os diferentes paradigmas de modelagem em NLP.

Em seguida, são descritos os principais tipos de modelos computacionais aplicados — desde os estatísticos, como TF-IDF e RAKE, até os modelos neurais modernos como Bidirectional Encoder Representations for Transformers (BERT) e LLM — assim como os algoritmos de classificação utilizados na pesquisa, com destaque para sua aplicabilidade na detecção de mensagens suspeitas em contextos forenses.

2.1 Conceitos de NLP

Devido a natureza técnica dentro do campo da Processamento de Linguagem Natural (NLP), emerge a necessidade de formalização de certos termos e conceitos abordados nesta pesquisa.

Esta investigação está assentada sobre o arcabouço da análise textual, com foco especial no processamento de linguagem natural. Assim, se faz necessário discernir as distinções presentes no domínio do NLP, que possam soar como sinônimos ao público em geral.

Objeto central da computação linguística, o texto, qual é formalmente entendido como um conjunto de palavras requer divisão em partes menores que seu processamento seja viável (Priberam, 2024b).

Neste ponto, convém que compreender exatamente como cada uma dessas partes são identificadas, aqui, seguindo a conceitualização proposta na modelagem do BERT, arquitetura sobre a qual se concentra este projeto.

2.1.1 Texto

Segundo Devlin et al. (2019), uma **sentença** é entendida de maneira distinta de uma frase, sendo uma extensão contígua de texto sem haver a necessidade de um sentindo linguístico real. Notória distinção conceitual pode ser observada quanto ao sentido linguístico de **frase**, qual seja uma unidade gramatical autônoma, capaz de transmitir um sentido completo em si mesma, podendo ser formada por uma ou múltiplas palavras, de acordo com Priberam (2024a).

Outro importante conceito é o de **sequência**, que trata de uma sequência de *tokens*. Esta pode ser composta por uma única sentença ou por várias sentenças

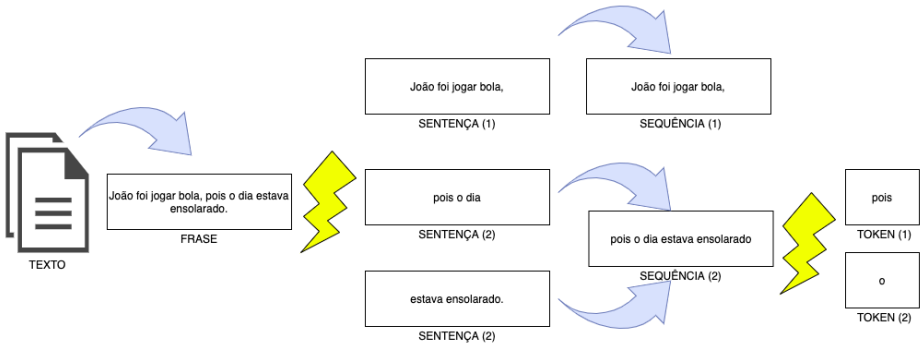
concatenadas.

Já o termo ***token*** refere-se a uma sequência de caracteres de tamanho variável que pode corresponder a uma palavra inteira, ou parte dela, a uma subpalavra (*subtoken*), conforme definido por Souza (2020).

A fim de elucidar tais conceitos, a Figura 2.1 ilustra como essas estruturas linguísticas são constituídas. Nela, é apresentado o texto como um conjunto de frases, que são elementos com significado completo e compreensível de forma independente, sem necessidade de outros contextos. Em seguida, uma frase podendo ser subdividida em sentenças, unidades semânticas menores, de acordo com a pontuação e gramática.

Por outro lado, as sequências se distinguem das sentenças, pois sua composição não está diretamente vinculada à semântica ou à gramática. Elas são formadas por *tokens* contíguos, que correspondem à representação mínima de elementos textuais, como palavras, sinais de pontuação ou números.

Figura 2.1: Conceitos em NLP



Fonte: Elaborado pelo autor.

Uma vez compreendida essa partição textual, tem-se outro elemento fundamental para o processamento de linguagem: os ***embeddings***. Estes representam os *token* de forma vetorial, (SOUZA, 2020). Cada vetor encapsula valores que expressam as características dessa palavra (*token*). Esta é uma importante estrutura dentro da NLP, pois é a partir dela que a parte léxica pode ser compreendida pelos modelos, devido essa conversão (representação) da palavra em um vetor numérico.

2.1.2 Modelagem

Para que seja possível a aplicação dessas estruturas citadas acima, faz-se necessário o emprego de modelos de Processamento de Linguagem Natural (NLP), que são algoritmos que partem de diferentes paradigmas e que são capazes de gerar textos, realizar inferência textual ou compreender contexto, possibilitando diversas aplicações.

Dentro dessas possíveis aplicações encontram-se algumas abordadas neste trabalho: extração de palavras e frases-chave e classificação de sentenças.

Por **palavras-chave** entende-se linguisticamente como a representação sintética do conteúdo informacional contido no documento em análise, como uma forma de indexação (NAVES, 2001), capaz de encapsular sua essência e permitir deduções relativas ao texto analisado.

Semelhante entendimento é mantido na aplicação de técnicas de extração de palavras-chave computacionalmente, apenas com a mudança de que sua execução não é realizada por pessoas, quer pelo próprio autor, ou leitor, mas por modelos de NLP.

Esse emprego da computação tem o intento de proporcionar maior agilidade na execução dessa tarefa, especialmente em se tratando de grande quantidade de textos que abrangem uma diversidade de temas, como é comum ocorrer no caso de conversas em aplicativos de mensagem.

Vale destacar que a extração de **frases-chave** possui a mesma construção cognitiva, exceto pelo tamanho máximo (número de palavras conexas) que o agente arbitra em sua execução.

Diferentemente dos modelos apresentados acima que buscam extrair a ideia central de um texto analisado, a **classificação de sentenças** (*sentence classification*) tem por objetivo rotular as sentenças como pertencentes a um tipo especifico de classe, conforme Minaee et al. (2020), podendo ser aplicada em diferentes estruturas textuais, como: frases, consultas, parágrafos e documentos:

Essa abordagem possui dois grandes agrupamentos de modelos, organizados a partir de sua arquitetura.

- baseado em regras: classificam o texto utilizando um conjunto de regras pré-definidas, exigindo profundo conhecimento do tema;
- baseado em *ML*: utilizam a observação dos dados, identificando padrões para que os textos possam ser classificados.

Destaca-se que esta pesquisa se deterá aos modelos de ML em seus experimentos, visto que se trata do enfoque deste trabalho o emprego do técnicas de inteligência artificial no auxílio das atividades de perícia forense.

A título ilustrativo, pode-se perceber os diferentes resultados hipotéticos obtidos a partir da aplicação desses diferentes modelos.

Texto analisado: *"Ontem João queria jogar bola, e para isso ele precisava de pelo menos um amigo, porém nenhum de seus amigos podia, pois tinham que fazer tarefa."*

- **palavras-chave**: bola, amigo, tarefa, jogar
- **frases-chave (*bigram*)**: jogar bola, seus amigos, fazer tarefa
- **classificação de sentenças**

Sentença	**Classe**
Ontem João queria jogar bola,	esporte
e para isso ele precisava de pelo menos um amigo,	amizade
porém nenhum de seus amigos podia,	amizade
pois tinham que fazer tarefa.	estudo

Como mencionado anteriormente, os modelos de análise linguística se fundamentam em certos paradigmas, dentre eles os modelos pré-treinados. Nesse tipo de abordagem, o modelo é inicialmente treinado em uma ampla coleção textual (*corpus*) com o objetivo de adquirir conhecimento geral sobre a língua, como vocabulário, estruturas sintáticas e padrões semânticos.

A partir desse conhecimento generalista, o modelo pode ser posteriormente ajustado para aplicações específicas por meio de técnicas de ajuste (*adaptation techniques*), como o *fine-tuning*, que consiste em continuar o treinamento do modelo utilizando um conjunto de dados específico da tarefa em questão.

Nesta etapa inicial, denominada fase generalista, destaca-se o conceito de ***corpus***, que, segundo Mendes (2016), consiste em um conjunto de textos escritos ou de transcrições de registros orais.

Esse grande conjunto textual é utilizado para passar conhecimento ao modelo, como no caso do BERTimbau que utilizou um *corpus* de 2,68 bilhões de *tokens* para que o modelo fosse capaz de compreender a língua portuguesa falada no Brasil, segundo Souza (2020).

2.2 Estrutura de Modelos Computacionais de NLP

Nesta seção apresenta-se o embasamento teórico-conceitual dos modelos e técnicas de Processamento de Linguagem Natural (NLP) considerados na pesquisa. O objetivo é traçar um panorama do tratamento computacional de textos, abordando não apenas suas aplicações práticas, mas também as bases e arquiteturas que sustentam a execução dos algoritmos e as propostas originais de seus autores.

Busca-se, assim, delinear a evolução do processamento de linguagem natural, destacando como determinados modelos e arquiteturas se tornaram estruturantes para a área, a exemplo dos *Transformers*, que passaram a constituir o núcleo dos modelos de estado da arte.

Convém ressaltar que cada modelo tem sua aplicação a depender da necessidade e da capacidade de processamento. Um exemplo engloba os modelos estatísticos, e.g., RAKE (ROSE et al., 2010) e TextRank (MIHALCEA; TARAU, 2004); que, em geral, não necessitam de pré-treinamento, dependendo única e exclusivamente do texto-objeto. Por outro lado, modelos baseados em redes neurais normalmente requerem bases de dados grandes ou então podem ser ajustados a partir de um treinamento prévio em outra base de dados.

Este é o caso de modelos BERT (DEVLIN et al., 2019) e os *Large Language Models* (LLM), que possuem maior capacidade de compreensão contextual além da possibilidade da aplicação de *fine-tuning* com intuito de especializar seus modelos originais à um dedicado ao caso em questão.

Todavia, este processo depende da viabilidade de geração de uma base específica para tal treinamento, bem como de grande capacidade computacional de acordo com modelo aplicado.

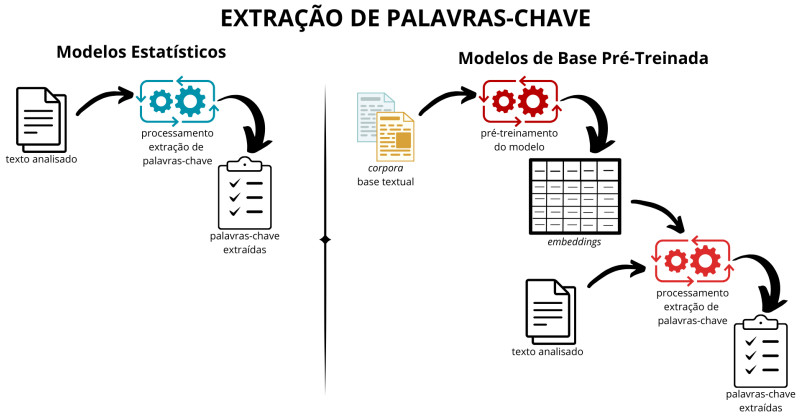
A Figura 2.2, ilustra uma distinção fundamental entre os modelos estatísticos de NLP e os modelos pré-treinados, a partir da aplicação da técnica de extração de palavras-chave. Em suma, os primeiros operam diretamente sobre o texto objeto para obtenção das palavras-chave, em etapas intermediárias.

Por outro lado, o segundo grupo possui uma quantidade maior e mais complexa de etapas. Inicialmente, eles são treinados em uma extensa coleção textual, conhecida como *corpora*, quando são gerados os *embeddings* do vocabulário, sendo que os *embeddings* são representações vetoriais numéricas de características de cada palavra.

Posteriormente, esses vetores, que compõe o modelo pré-treinado, são uti-

lizados pelo modelo de extração para analisar o texto objeto considerando os pesos pré-calculados e assim obter as palavras-chave.

Figura 2.2: Diferença na estrutura entre modelos de NLP estatísticos e de base pré-treinada



Fonte: Elaborado pelo autor.

2.3 Modelos de Extração de Frases-Chave

Nesta secção discorremos sobre como os modelos de NLP tem sido objeto de pesquisa e investigação no meio acadêmico. Analisamos recentes pesquisas e soluções encontradas para sua aplicação na resolução de problemas da vida real. Também examinamos a evolução do seu desenvolvimento e as técnicas inovadoras aplicadas a fim de otimizar seu desempenho.

Os modelos são categorizados a partir do conceito fundamental que norteia sua construção. Ademais, cabe salientar que tal categorização, aqui proposta, é estruturada de forma a elucidar a ideia das etapas de seu desenvolvimento ao longo dos anos.

2.3.1 Modelos Estatísticos de Linguagem (SLM)

Uma das primeiras classes de modelos de linguagem natural, os SLM, tiveram seu início nos anos 90, de acordo com Zhao et al. (2023), tiveram grande relevância no desenvolvimento desta área da computação, bem como trazendo a possibilidade de aplicação de tecnologias linguísticas.

Segundo Rosenfeld (2000), algumas dessas aplicações iniciais foram correção ortográfica, reconhecimento de fala, tradução automática, reconhecimento de caracteres, classificação de documentos, entre outros.

Os SLM possuem grande destaque entre os modelos de linguagem natural, visto que, devido seu caráter estatístico que busca gerar informações a partir de cálculos dentro do próprio documento objeto, muitos deles podem ser aplicados em diversos idiomas sem a necessidade que qualquer adaptação.

Logo, dada sua representatividade, seu poder informacional e, em geral, seu potencial de processamento, detalhamos na sequência alguns desses modelos existentes.

2.3.2 *Term-Frequency Inverse Document Frequency* (TF-IDF)

Conforme Erra et al. (2015), esse modelo calcula a importância de uma palavra em um documento, considerando a quantidade de vezes que essa se repete e a proporção inversa com que essa mesma palavra aparece no conjunto de documentos outros documentos.

Assim, primeiramente o modelo calcula a frequência de uma palavra no documento, conforme Equação 2.1:

$$\mathrm{tf}(t, d) = \frac{\mathrm{f}(t, d)}{|d|} \tag{2.1}$$

onde:

- $\mathrm{tf}(t, d)$: frequência do termo t no documento d (normalizada);
- t: termo (palavra/token) de interesse;
- d: documento (texto analisado);
- $f(t, d)$: número de ocorrências de t em d (frequência absoluta);
- $|d|$: tamanho de d, medido como o total de termos (tokens/palavras) no documento.

Na sequência é obtida a frequência inversa do documentos, que consiste em penalizar palavras mais comuns nos documentos, como demonstrado na Equação 2.2:

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d | t \in d\}|} \tag{2.2}$$

onde:

- $\text{idf}(t, D)$: frequência inversa de documentos (inverse document frequency) do termo t na coleção D;
- t: termo (palavra/token) de interesse;
- D: coleção (corpus) de documentos;
- $|D|$: número total de documentos na coleção D;
- $\{d \in D \mid t \in d\}$: conjunto de documentos d da coleção D que contêm o termo t;
- $|\{d \in D \mid t \in d\}|$: quantidade de documentos em D nos quais o termo t ocorre.

Finalmente os dois resultados são combinados, tal qual apresentado abaixo na Equação 2.3:

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D) \tag{2.3}$$

onde:

- $\text{tf-idf}(t, d, D)$: peso *tf-idf* do termo t no documento d, considerando a coleção D;
- t: termo (palavra/token) de interesse;
- d: documento (texto analisado);
- D: coleção (corpus) de documentos;
- $\text{tf}(t, d)$: frequência do termo t no documento d (normalizada), conforme a Equação 2.1;
- $\text{idf}(t, D)$: frequência inversa de documentos do termo t na coleção D, conforme a Equação 2.2;
- $\times$: operador de multiplicação, combinando relevância local (tf) e raridade na coleção (idf).

Logo, intuitivamente é possível perceber que são considerados mais relevantes os termos que apareçam várias vezes num documento, porém não em muitos documentos da coleção.

2.3.3 *Rapid Automated Keyword Extraction* (RAKE)

Modelo de extração de palavras-chave que baseia seu cálculo com base na frequência e co-ocorrência das palavras.

De acordo com Rose et al. (2010), o modelo RAKE parte de dois princípios: a) palavras-chave frequentemente possui múltiplas palavras e b) raramente são acompanhadas de pontuação ou palavras de parada (*stop-words*).

O modelo consiste em separar o texto em um conjunto de palavras-chave candidatas, que são sequências de palavras de conteúdo (cujo significado léxico não seja mínimo) da forma que elas ocorrem no texto. Para isso, são usadas as palavras de paradas e os delimitadores de frases fornecidos na entrada de dados.

Na sequência RAKE gera a matriz de coocorrências, que consiste em um grafo, que associa cada termo de uma palavra-chave candidata com outros termos de uma palavra-chave candidata. Quando a matriz de co-ocorrência está completa, é calculado uma pontuação para cada termo de uma palavra-chave candidata, assim como disposto na Equação 2.4, que consiste na proporção entre grau (deg(w)) (número de arestas de cada palavra – vértice – do grafo) e frequência (freq(w)).

$$\frac{\deg(w)}{\text{freq}(w)} \tag{2.4}$$

onde:

- w: uma palavra (termo) candidata extraída do texto;
- freq(w): frequência de ocorrência de w no texto (número de vezes em que w aparece);
- deg(w): grau (*degree*) de w, isto é, o total de coocorrências de w com outras palavras dentro das frases candidatas do RAKE (em geral, soma dos tamanhos das frases candidatas em que w aparece, contabilizando os demais termos).

Consequentemente, é possível trabalhar o conceito de *n-gram* com esse modelo, visto que uma vez definido o valor do *n-gram*, o modelo encontra as palavras com maior representatividade somando os valores das proporções de cada termo, isso para as palavras adjacentes.

2.3.4 TextRank

Modelo baseado em uma estrutura de grafo, inspirado no algoritmo *PageRank*, que busca identificar as palavras-chave de um texto a partir de um sistema de "votação" ou "recomendação".

TextRank, segundo Mihalcea & Tarau (2004), considera que quando um vértice se liga a outro, ele estaria pontuando para esse outro vértice (nó). Assim, quanto maior o número de votos – ligação – que um vértice recebe, maior seria sua relevância no texto. O modelo não avalia apenas os votos do nó, mas também a importância do deste nó.

Isso posto, temos abaixo o cálculo da pontuação do vértice, através da Equação 2.5:

$$\mathrm{WS}(V_i) = (1-d) + d \times \sum_{V_j \in \mathrm{In}(V_i)} \frac{w_{ji}}{\sum_{V_k \in \mathrm{Out}(V_j)} w_{jk}} \mathrm{WS}(V_j) \tag{2.5}$$

- $\mathrm{WS}(V_i)$: pontuação do vértice V_i;
- d: fator de amortecimento (*damping*), sendo uma constante que garanta uma pontuação miníma ao vértice (definida como 0.85 pelo modelo);
- $\sum_{V_j \in \mathrm{In}(V_i)}$: somatória dos vértices V_j que estejam apontados para V_i;
- $\frac{w_{ji}}{\sum_{V_k \in \mathrm{Out}(V_j)} w_{jk}}$: peso da aresta entre V_i e V_j dividido pelo somatório dos pesos das arestas saindo de V_j; e
- $\mathrm{WS}(V_j)$: pontuação atual do vértice V_j.

O TextRank inicia seu processamento atribuindo um valor arbitrário a cada vértice (palavra). Em seguida, a fórmula apresentada é aplicada de forma iterativa até que os valores dos nós convirjam. Isso significa que, a cada iteração, a alteração nos valores torna-se cada vez menor, até que não seja mais possível identificar diferença relevante em relação ao valor real.

Por fim, os *tokens* são ranqueados com base em sua pontuação final.

2.4 Modelos Classificadores

Conjunto de algoritmos de *Machine Learning* pertencente ao grupo de aprendizado supervisionado, que segundo James et al. (2013) tem objetivo de prever

uma saída – rótulo – com base em uma ou mais entradas, diferentemente dos modelos não supervisionados que buscam identificar relações e estruturas entre os dados.

A ideia central no conceito de supervisionado, vem de que cada amostra x_i, com suas respectivas características, possui uma resposta y_i.

Cabe ressaltar que os modelos de aprendizado de máquina podem ser divididos em duas categorias principais: discriminativos e generativos.

A abordagem discriminativa, amplamente utilizada para tarefas de classificação, tem como objetivo principal encontrar a fronteira de decisão que melhor separa as classes. Para isso, ela busca a probabilidade de uma classe y, dada uma entrada x, através da probabilidade condicional (SAMMUT; WEBB, 2010).

Já a categoria generativa adota uma abordagem diferente. Em vez de focar na fronteira de decisão, esses modelos aprendem a distribuição dos dados para cada classe, o que lhes permite não apenas classificar, mas também gerar novas instâncias de dados com base nas classes existentes.

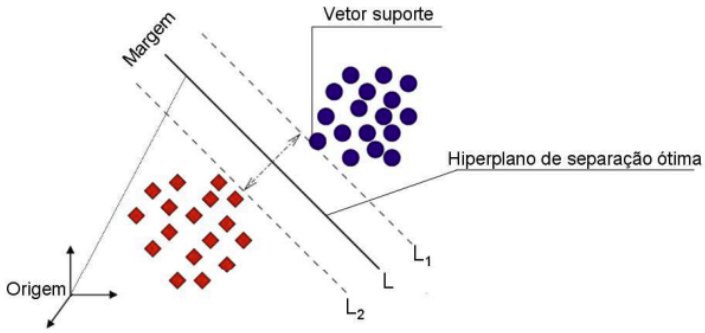
Assim, presente seção apresenta a conceitualização dos modelos de classificação utilizados na pesquisa, a saber:

- *Support Vector Machine* (SVM);
- *Multilayer Perceptron* (MLP);
- Random-Forest;
- XGBoost;
- Naive Bayes.

2.4.1 SVM

Modelo baseado em cálculos estatísticos, Vapnik (1999), podendo ser utilizado tanto como classificador como regressor, conceitualmente podendo ser compreendido como tendo por princípio a criação de um hiperplano (L) que maximize a distância entre as classes de maneira mais evidente – por meio de suas margens L_1 e L_2 – logo uma minimização de erros, assim como demonstrado na Figura 2.3.

Figura 2.3: Exemplo de classificação com SVM



Fonte: Nascimento et al. (2009).

Conforme Nascimento et al. (2009), o algoritmo possui a seguinte modelagem: dado um conjunto de amostras de treinamento x_i, y_i, onde $\mathbf{x}_i \in \mathbb{R}^M$ como sendo os vetores de característica, e $y_i \in \{-1, 1\}$ a respectiva classe, ou rótulo, visto que se trata de um modelo supervisionado; ele passa por um processo de treinamento onde possa aprender por meio dessas amostras de tal forma que seja capaz de resolver a classificação de um novo exemplo (x, y) ainda não conhecido, dada a distribuição de probabilidade aprendida na fase de treinamento.

O objetivo de seu processo é a minimização da expectativa de erro $\epsilon(\zeta)$ da classificação, por meio da Equação 2.6:

$$\epsilon(\zeta) = \int \frac{1}{2} \left| y - \mathrm{f}(x, \zeta) \right| d\mathrm{P}(x, y) \tag{2.6}$$

onde:

- $\left| y - \mathrm{f}(x, \zeta) \right|$ é erro absoluto dado pelo valor real de y e a função de decisão $\mathrm{f}(x, \zeta)$; e
- $d\mathrm{P}(x, y)$ é distribuição de probabilidade.

Todavia, ainda segundo Nascimento et al. (2009), essa distribuição de probabilidade real não é necessariamente conhecida, assim, a Equação 2.6 necessita ser ajustada para um risco empírico, qual seja:

$$\epsilon_\psi(\zeta) = \frac{1}{2D} \sum_{i=1}^{D} \left| y_i - \mathrm{f}(x_i, \zeta) \right| \tag{2.7}$$

onde:

- D: tamanho da amostra;
- $\frac{1}{2D}$:fator de normalização para cálculo do erro médio;
- $\sum_{i=1}^{D} \left| y_i - f(x_i, \zeta) \right|$: somatório dos erros de todos as amostras de treinamento;
  - $\left| y_i - f(x_i, \zeta) \right|$: erro absoluto para determinada amostra i.

Ou seja, é realizado o somatório dos erros dos elementos de treinamento e normalizado pelo dobro do tamanho da amostragem, levando em consideração o tamanho do conjunto.

2.4.2 *Naive Bayes*

Outro modelo de base estatística, tendo como ponto de partida o Teorema de Bayes, recebe seu nome como ingênuo (*naive*) dado seu pressuposto que, conforme Oguri (2006), todos seu atributos sejam independentes, logo, o valor de uma característica não impactaria em outra.

Ainda conforme o autor, esse teorema foi um dos primeiros a inverter a lógica comumente utilizada pela estatística da época, século XVIII, conhecida como *forward probability*, onde a partir de determinadas condições, haveria certa probabilidade de ocorrência de um evento.

Todavia, em seu teorema, Thomas Bayes busca, a partir dos eventos observados identificar quais condições possibilitaram sua ocorrência, chegando a Equação 2.8 de probabilidade condicionada (KUBAT, 2017).

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})} \tag{2.8}$$

onde:

- $P(\mathbf{x}|c_i)$: probabilidade de x dado c;
- $P(c_i)$: probabilidade *a priori* de c;
- $P(\mathbf{x})$: densidade de probabilidade

Dessa maneira, ao considerar um cenário de vários atributos – por pressuposto considerados como independentes – o resultado pode ser expressado pelo produtório das probabilidades disposto na Equação 2.9:

$$P(\mathbf{x}|c_j) = \prod_{i=1}^{n} P(x_i|c_j) \tag{2.9}$$

Conforme Kubat (2017), a classificação do elemento em análise se dá pela classe que maximizar o valor bayesiano, conforme Equação 2.10:

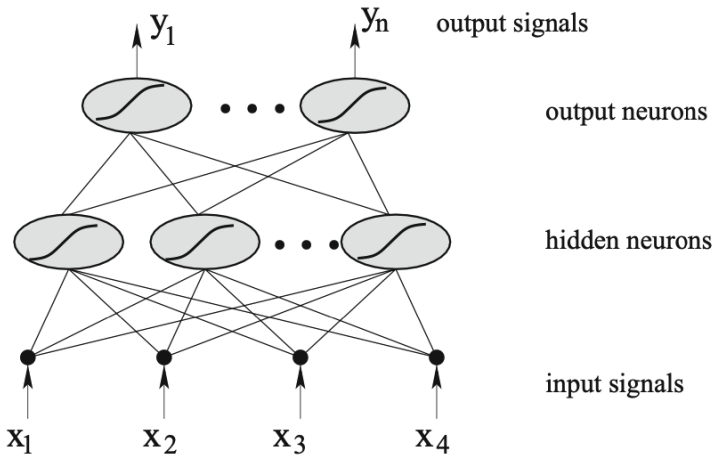
$$P(c_j) \cdot \prod_{i=1}^{n} P(x_i|c_j) \tag{2.10}$$

2.4.3 MLP

A Multilayer Perceptron (MLP), ou Perceptron Multicamadas, é um modelo pertencente à classe das redes neurais *feedforward*. Como o próprio nome indica, trata-se de uma rede composta por múltiplos perceptrons organizados em camadas. O perceptron, constitui a unidade fundamental de processamento, capaz de receber um conjunto de entradas, ponderá-las por meio de pesos, somá-las e aplicar uma função de ativação para gerar uma saída.

Modelo pertencente à classe das redes neurais *feedforward*, qual consiste, secundo (GARDNER; DORLING, 1998), numa estrutura composta por camadas, a saber: uma de entrada, uma ou mais camadas ocultas e uma camada de saída, conforme Figura 2.4.

Figura 2.4: Exemplo de uma rede neural composta por duas camadas ocultas.



Fonte: Kubat (2017).

Na Figura 2.4, temos nos símbolos de formato oval a representação dos neurônios, sendo a unidade básica da MLP, onde vê-se sua total interconexão com os neurônios de outras camadas, porém sem comunicação com os localizados na mesma *layer*, Kubat (2017).

De acordo com o autor, o modelo recebe os valores dos atributos, que são passados aos neurônios pelos respectivos link – que possuem um peso para cada ligação. Assim, o neurônio tem a função de receber esses valores, uma vez multiplicados pelos pesos, somá-los e gerar uma saída por meio de uma função de ativação, no caso, uma sigmoide.

A segunda camada, proposta no exemplo, repete o mesmo processo, gerando uma saída considerando o resultado das operações realizadas pelas duas camadas, como na Equação 2.11:

$$y_i = \mathrm{f}\left(\sum_j w_{ji}\mathrm{f}\left(\sum_k w_{kj}x_k\right)\right) \tag{2.11}$$

- x_k: valores dos atributos;
- $\sum_k w_{kj}$: somatório dos valores de entrada, atributos, ponderados pelos pesos das associações w_{kj}, primeira camada oculta;
- $\sum_j w_{ji}$: somatório do resultado do processamento da segunda camada oculta, dados os pesos w_{ji};

- f: função de ativação.

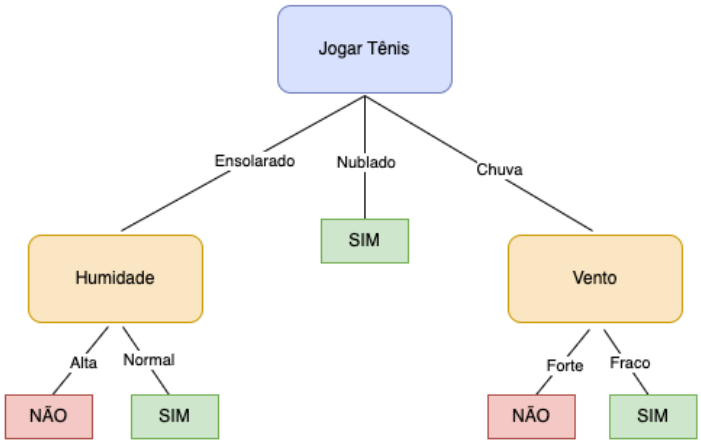
Assim, após o processamento de todas as camadas a atribuição da classe é feita considerando o valor mais alto dos neurônios de saída.

2.4.4 *Random Forest*

Algoritmo proposto por Breiman (2001), que tem como base o modelo de classificação Árvore de Decisão (*Decision Tree*), daí seu nome Floresta Aleatória (*Random Forest*), uma vez que executa sua previsão a partir da geração de várias árvores de decisão, (BRITO, 2019).

Em sua modelagem, as árvores de decisão classificam um elemento iniciando por um atributo "raiz", e aplicando testes com relação às características, em cada "nó", até finalizar a escolha (MITCHELL, 1997) como apresentado na Figura 2.5:

Figura 2.5: Exemplo de uma *Decision Tree* para escolha sobre jogar tênis.

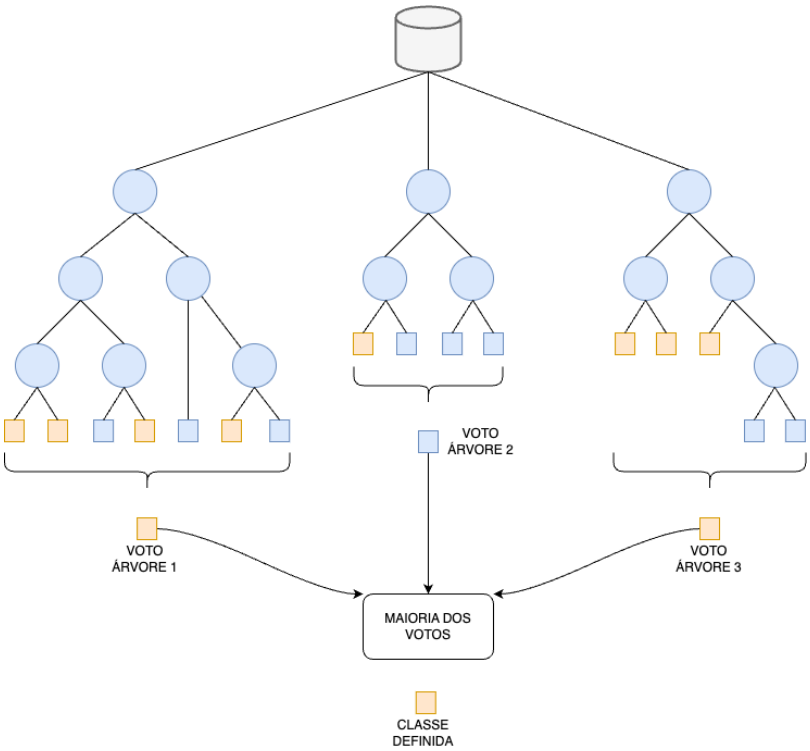


Fonte: Elaborado pelo autor, adaptado de Mitchell (1997).

Na construção das árvores, cada uma é treinada a partir de subconjuntos de dados de treinamento gerados aleatoriamente (HO, 1995). Além disso, em cada nó de decisão ocorre também uma seleção aleatória de subconjuntos de atributos, o que introduz maior diversidade entre as árvores e reduz a correlação entre elas.

Uma vez treinadas, a predição de classe pelo *Random Forest* se dá pelo voto majoritário, ou seja, a classe escolhida pela maioria das árvores é definida como resultado final da classificação (LACERDA, 2023), como ilustrado na Figura 2.6.

Figura 2.6: Exemplo de definição de classe pelo *Random Forest*.



Fonte: Elaborado pelo autor, adaptado de Mourão (2022).

2.4.5 XGBoost

Modelo baseado em árvores de decisão, no caso, uma floresta, nisto se assemelhando ao modelo anterior: *Random Forest*.

Segundo Rocha (2018), trata-se de uma melhoria do algoritmo de *gradient boosting*, sendo o XGBoost um modelo iterativo tendo como princípio a melhoria, a cada iteração, do modelo de árvore anterior.

Em linhas gerais, cada árvore construída no XGBoost atribui uma pontuação a uma possível classe. Essas pontuações são então somadas, produzindo a previsão final do modelo. Esse processo aditivo permite que o algoritmo aprenda de maneira incremental, ajustando-se progressivamente a partir dos erros identificados em etapas anteriores.

Esse acaba sendo um dos principais diferenciais em relação ao *Random Forest*: enquanto este gera múltiplas árvores de forma independente, o XGBoost adota um processo iterativo, no qual cada nova árvore é construída para corrigir os erros dos modelos anteriores. Dessa forma, o algoritmo ajusta-se progressivamente, refinando suas previsões a cada etapa.

Outro aspecto importante é a utilização de mecanismos de regularização. Esses mecanismos controlam a complexidade das árvores e reduzem o risco de *overfitting*, garantindo melhor capacidade de generalização.

Por meio dessa combinação — aprendizado iterativo, soma de contribuições das árvores e regularização — o XGBoost pode a alcançar um alto desempenho em tarefas de classificação e regressão.

2.5 Modelos de Linguagem Natural baseados em Redes Neurais

O desenvolvimento de técnicas de NLP passou, inicialmente, por modelos estatísticos tradicionais, que se mostraram eficazes em tarefas mais simples, como a contagem de frequências e a identificação de palavras-chave. Com a evolução das pesquisas, entretanto, surgiu a necessidade de abordagens mais sofisticadas capazes de lidar com dependências de longo alcance e com a complexidade semântica das línguas naturais. Nesse contexto, os modelos baseados em redes neurais ganharam destaque, trazendo contribuições significativas ao possibilitar a previsão de sequências de palavras Zhao et al. (2023), além da introdução de representações distribuídas que capturam relações mais profundas entre termos. Esses elementos tiveram impacto decisivo no avanço do processamento textual, consolidando as redes neurais como uma das principais bases para os modelos atuais.

2.5.1 *Recurrent Neural Networks* (RNN)

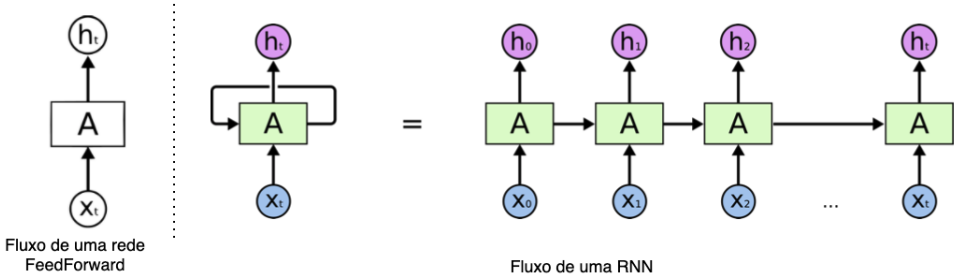
Modelo criado com potencial de lidar com informações sequenciais. Neste sentido, tem-se que a *Recurrent Neural Networks* (RNN) (rede neural recorrente) utilizam variáveis de estado a fim de guardar informações anteriores e processam conjuntamente com as entradas presentes para calcular as saídas atuais (ZHANG et al., 2021) ou seja, a predição léxica atual depende dos cálculos

realizados anteriormente.

De acordo com Luz (2019), diferentemente das redes *feedforward*, a RNN possui conexões entre suas unidades internas, o que permite uma simulação temporal dinâmica, sendo capazes de utilizar sua memória interna para processar sequências de entrada.

Como pode ser observado na Figura 2.7, o fluxo da rede RNN possui essa capacidade de analisar sequência de dados – incluindo textos – que se dá por sua arquitetura de *loop*, a qual permite que informações anteriores possam influenciar a presente saída da rede, onde as unidades posteriores além de receberem um novo *input* externo, recebem outra da unidade anterior.

Figura 2.7: Comparação entre os fluxos rede *FeedForward* e *RNN*



Fonte: Elaborado pelo autor, adaptado de Luz (2019).

Esse potencial permite que as RNN possam ser aplicadas com objetivo de modelar a probabilidade de uma sequência de dados, como no caso de tarefas relacionadas ao Processamento de Linguagem Natural.

Diferentemente dos modelos n-gramas que calculam a probabilidade da próxima palavra com base nas n palavras anterior, a rede recorrente utiliza todas as palavras anteriores no cálculo da probabilidade da próxima, Bengio et al. (2003).

$$\hat{P}(\mathbf{w}_1^T) = \prod_{t=1}^{T} \hat{P}(\mathbf{w}_t | \mathbf{w}_1^{t-1}) \tag{2.12}$$

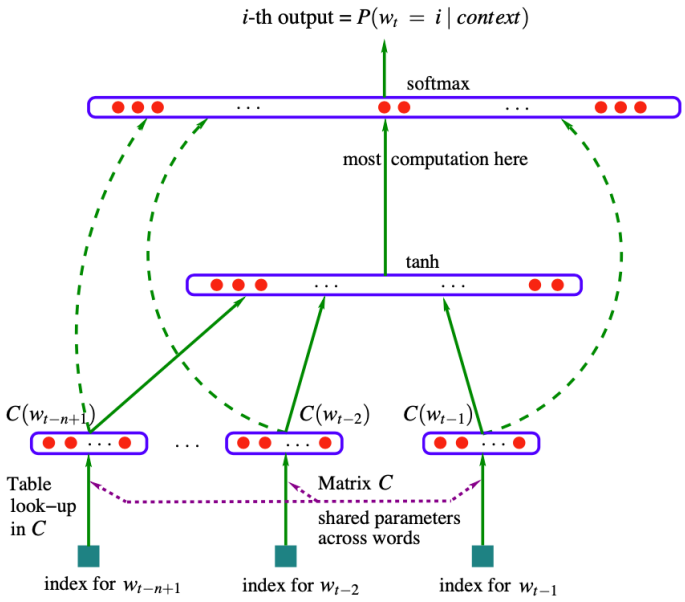
Assim, vê-se que a probabilidade de toda a sequência (Equação 2.12), da primeira até a T-ésima, se dá pelo produtório das probabilidades condicionais da palavra $\mathbf{w}_t$ considerando todas as anteriores $\mathbf{w}_1^{t-1}$.

Bengio et al. (2003) apresenta um modelo que busca aprender a representação

de probabilidades de sequência de palavras de um vocabulário grande, porém finito.

O modelo, conforme Figura 2.8, faz um mapeamento desse vocabulário – mapeamento C – onde cada palavra é representada por um vetor de características (vetor de *embeddings*) gerando uma matriz $|V| \times m$, onde $|V|$ é o tamanho do vocabulário e m a dimensão do vetor.

Figura 2.8: Arquitetura neural



Fonte: Bengio et al. (2003).

A partir do vetores gerados, uma função g (rede neural) calcula a probabilidade condicional da próxima palavra de uma sequência, dadas as anteriores.

Na sequência, esses valores são normalizados passando por uma função *softmax*, descrita na Equação 2.13, para prever a próxima palavra da sequência. Essa função tem por objetivo de garantir que não haja valores negativos, e que sua soma seja igual a 1, demonstrando assim suas respectivas probabilidades (ZHANG et al., 2021).

$$y_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)} \tag{2.13}$$

Dessa maneira o modelo estima a probabilidade da próxima palavra (w_t)

levando com consideração as palavras anteriores, permitindo a geração de uma sentença coerente e contextualizada.

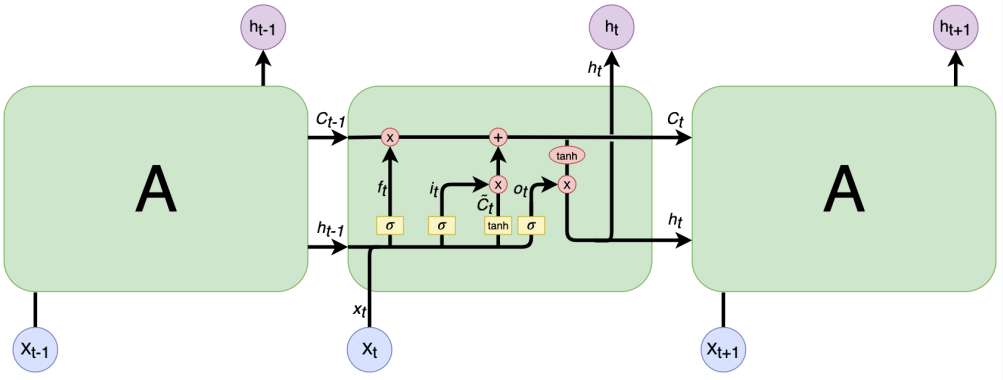
Apesar do grande potencial das redes neurais, estudos como o de Luz (2019) apontam que arquiteturas cujo treinamento depende do gradiente descendente (*gradient descent*) e da retropropagação (*backpropagation*), como as RNN, enfrentam o problema do gradiente desvanecente.

Nesse fenômeno, o gradiente responsável por ajustar os pesos da rede torna-se progressivamente menor a cada iteração do *backpropagation*, o que compromete o aprendizado e dificulta a captura de dependências de longo prazo.

2.5.2 *Long Short-Term Memory* (LSTM)

Uma derivação da RNN, a LSTM possuem a capacidade de lidar com dependências de longo prazo (LUZ, 2019) visto que foram projetadas para resolver o problema do gradiente desvanecente sofrido pelas redes recorrentes, como abordado na subseção acima. A inovação deste modelo é sua célula de memória, a qual utiliza de portões especiais para controlar o fluxo informacional, representada na Figura 2.9, pela linha superior horizontal na unidade central, a qual recebe o estado da célula anterior C_{t-1} e gera o novo estado C_t.

Figura 2.9: Fluxo e portões da rede LSTM



Fonte: Elaborado pelo autor, adaptado de Luz (2019).

Segundo Luz (2019), a dinâmica do LSTM é moldada pelo funcionamento dos portões, estruturas que possuem finalidades específicas, como: esquecimento (*forget gate* – f_t), entrada (*input gate* – i_t) e saída (*output gate* – o_t); quais permitem

que a rede decida quais informações manter, descartar ou modificar a cada instância.

Assim, a partir da esquematização apresentada na Figura 2.9 pode-se elucidar o funcionamento de uma célula LSTM.

Inicialmente, o **portão de esquecimento** (f_t) define quais informações do estado anterior (h_{t-1}, C_{t-1}) devem ser mantidas. Em seguida, o **portão de entrada** (i_t), em conjunto com a função de ativação *tanh*, determina as novas informações candidatas ($\tilde{C}_t$) a compor o estado da célula. A soma ponderada desses elementos gera o novo estado C_t.

Por fim, o **portão de saída** (o_t) regula quais partes de C_t serão usadas para produzir o novo estado oculto (h_t), aplicando novamente a função *tanh*.

A arquitetura da LSTM foi capaz de superar o desafio das dependências de longo prazo enfrentado pelas RNNs tradicionais. Isso devido sua inerente capacidade de avaliar e manter apenas as informações que julgar cruciais, descartando as irrelevantes, e consequentemente aprimorando o processo de aprendizado sequencial, uma vez que a própria rede determina o que é importante para seu aprendizado (LUZ, 2019).

2.5.3 Modelos de Linguagem Pré-Treinados

Pertencentes à classe de técnicas de NLP, os modelos de linguagem pré-treinados (*Pre-trained Language Model (PLM)*) expandiram significativamente o potencial de processamento linguístico. Esses modelos têm em comum a premissa de adquirir conhecimento por meio de um treinamento inicial realizado sobre extensos *corpora*, que servem como *input* para a construção de representações robustas da linguagem.

Dado o objetivo desta pesquisa de aplicar o modelo **Bidirectional Encoder Representations for Transformers (BERT)** (Representações de Codificador Bidirecional de Transformadores) como motor de análise textual, esta parte do trabalho enfoca o detalhamento desse modelo cujo grande diferencial está no seu poder de compreensão contextual devido sua arquitetura bidirecional – compreensão tanto da direita para esquerda, como da esquerda para a direita, conforme Devlin et al. (2019) – de processamento de texto.

Segundo Zhang et al. (2021), o BERT combina características do ELMo e do GPT. Assim como o ELMo, é capaz de codificar o contexto de forma bidirecional, capturando dependências mais complexas entre palavras; contudo, supera-o

por não depender de arquiteturas específicas para cada tarefa. Por outro lado, mantém o caráter generalista observado no GPT, ainda que este último opere apenas em uma direção do texto.

Zhang Zhang et al. (2021) destaca ainda que o BERT é construído exclusivamente a partir do codificador dos *Transformers*. Isso contrasta com a arquitetura original proposta para o transformador, que combina codificador e decodificador, bem como com modelos geradores de texto, como o Generative Pre-Training (GPT), que utilizam apenas a parte decodificadora.

Assim, dado que a construção do modelo BERT é fundamentada na arquitetura de *Transformers* (ZHAO et al., 2023), esses modelo são pré-treinados com uma base não rotulada, fornecendo grande potencial semântico para uso geral, o que de certa forma incentivou a aplicação da técnica de *fine-tuning*, onde pode ser refinado para atender uma tarefa específica.

A ideia de "transferência de conhecimento", partindo de uma base de um aprendizado preestabelecida para então aprimorá-la vistas à execução de determinado objetivo, fora inspirada na visão computacional. Tal abordagem já era realizada em casos como do ImageNet (DEVLIN et al., 2019).

Ainda segundo o autor, isso desenha, com uma visão geral, sua estrutura: um pré-treinamento e o ajuste fino onde cada tarefa específica (*downstream*) tem seu modelo ajustado para uma resolução específica, possuindo como base os mesmos parâmetros do pré-treinamento.

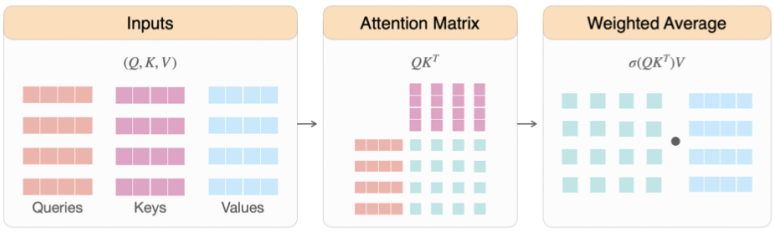
2.5.4 Transformer

Como mencionado acima, BERT é baseado na arquitetura dos *Transformers* (VASWANI et al., 2017), um modelo baseado exclusivamente em mecanismos de atenção, sem fazer uso de convoluções e recorrências, tendo o intuito de obter maior desempenho, além de permitir seu paralelismo e reduzir seu tempo de treinamento.

De acordo com Ribeiro (2023), o modelo então apresentado por Vaswani possui um mecanismo de atenção chamado de função de pontuação de atenção (*scaled-dot-product attention*), conforme demonstrado na Equação 2.14, sendo Q a consulta *Query*, K a chave *Key*, V o valor *Value* e σ a função *softmax* utilizada para converter os valores em probabilidades:

$$\text{attention}(Q, K, V) = \sigma\left(\frac{QK^T}{\sqrt{d_k}}V\right) \tag{2.14}$$

Figura 2.10: Função de pontuação de atenção *(scaled-dot-product attention)*



Fonte: Ribeiro (2023).

Na Figura 2.10, acima, notamos que QK^T representa o produto interno entre as matrizes Q e a transposta de K, gerando então uma matriz de atenção, demonstrando a similaridades entre os valores.

Já na última etapa, é aplicada a função *softmax*, utilizada para normalizar os valores numa distribuição de probabilidades.

Exemplificando, podemos partir de duas frases ilustrativas:

(a) "O muro de Berlim caiu."

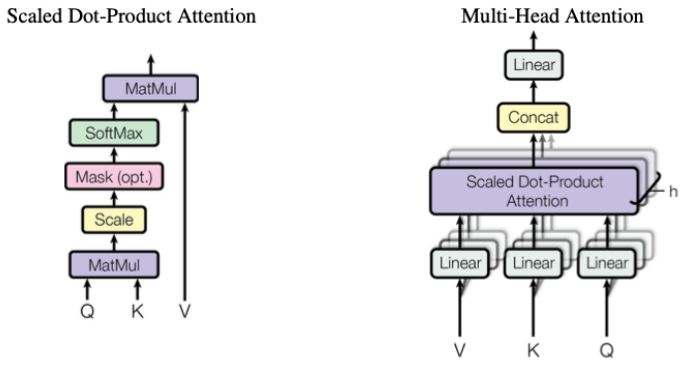
(b) "Caiu o muro de Berlim na prova."

Aqui o vocábulo "caiu" pode se referir tanto no sentido de queda, algo que tombou (a), como um conteúdo que fora cobrado em uma avaliação (b). Nesse sentido podemos vir a buscar identificar o contexto de "caiu" em uma das frases.

Assim, "caiu" seria a consulta Q desse exemplo, qual teria como objetivo o compreensão do contexto em que está inserida. As chaves K seriam as representações dos demais vocábulos contidos no textos, tais como: "muro", "Berlim" e "prova"; tais vetores são utilizados no desenho contextual ao qual a o termo "caiu" está envolvido, dada sua similaridade vetorial.

Outro importante conceito introduzido na arquitetura foi o *Multi-head*, onde, segundo Ribeiro (2023), os autores descobriram que o modelo poderia ser construído de forma a permitir o processamento em paralelo, calculando a atenção em separado, concatenando todos os resultados e projetando um resultado final, reduzindo o tempo de processamento durante a fase de treinamento, assim como ilustrado na Figura 2.11.

Figura 2.11: Produto escalar (esquerda) e *multi-head* (direita), funcionamento em paralelo

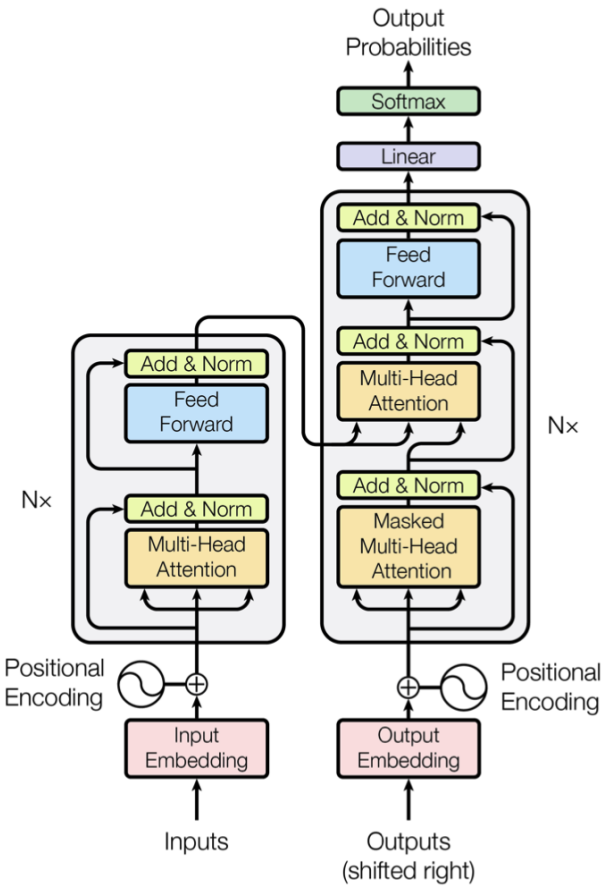


Fonte: Vaswani et al. (2017).

Assim como em outros modelos, os *Transformers* são compostos por duas estruturas, um codificador (*Encoder*) e um decodificador (*Decoder*) (VASWANI et al., 2017) como na Figura 2.12:

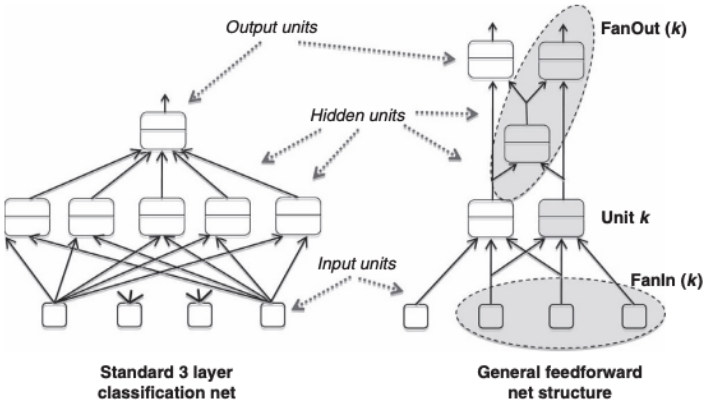
- *Encoder*: uma sequência de seis camadas idênticas empilhadas, cada qual possuindo duas subcamadas, *multi-head attention* e uma rede *feed forward*, seguido de uma camada de normalização após cada subcamada;
  - *feed forward*: tipo de rede unidirecional, acíclica – diferentemente das redes recorrentes – sendo estruturada, geralmente, em três camadas principais: entrada, oculta e saída, como demonstrado na Figura 2.13 (SAMMUT; WEBB, 2010);
- *Decoder*: composto também por seis camadas idênticas empilhadas – assim como no codificador, além das duas subcamadas iguais ao *encoder*, lhe é acrescentada uma terceira subcamada a qual aplica o *multi-head attention* sobre a saída do codificador, seguido de igual forma pela normalização; destaca-se a aplicação de mascaramento na subcamada de auto-atenção da pilha do decodificador.

Figura 2.12: Estrutura de composição dos *Transformers* – *Encoder* (esquerda) e *Decoder* (direita)



Fonte: Vaswani et al. (2017).

Figura 2.13: Estrutura de composição das redes *feed-forward*



Fonte: Sammut & Webb (2010).

Conforme demonstrado na Figura 2.12, o *Transformer* recebe como *input* um *Embedding* – que é uma representação vetorial do *Token* de entrada. Adicionalmente aos *Embeddings* de "*id*" são incorporados outros dois, um de sequência e outro de segmento (SOUZA, 2020). O primeiro visa marcar a posição de cada palavra dentro da entrada, uma vez que o modelo de *Transformer* – base do BERT – trabalha com conjunto de vetores, não identificando a ordem de sequência das palavras para seu processamento.

Já os de segmento, tem como objetivo identificar a qual sentença o respectivo *Token* pertence dentro de uma mesma sequência. Uma solução é muito importante para o modelo, visto que essa arquitetura proposta gera as representações de uma só vez, ao invés de fazê-lo sequencialmente como ocorre com as redes recorrentes.

Abaixo, na Tabela 2.1, um exemplo de como se dá tal entrada de dados utilizando a frase "João foi embora. Mas depois voltou":

Tabela 2.1: Exemplo de entrada de dados utilizando tokenizador BERTimbau

Token	[CLS]	João	foi	embora	.	[SEP]	Mas	depois	voltou	.	[SEP]
Token_id	101	1453	262	1853	119	102	1645	700	2927	119	102
Posição	1	2	3	4	5	6	7	8	9	10	11
Segmento	A	A	A	A	A	B	B	B	B	B	B

Fonte: Elaborado pelo autor, adaptado de Souza (2020).

Dentro da camada de atenção as palavras são avaliadas dado o contexto,

ou seja, as outras palavras contidas na sentença (RIBEIRO, 2023), de forma que palavras que possuam maior relevância para a palavra alvo tenham um peso maior.

Observa-se que devido o uso do *Multi-head*, serão geradas várias representações para a mesma palavra, sendo uma representação em cada *head*, cada um dando um enfoque diferente para a palavra. Ao final todas essas representações são concatenadas de forma a gerar uma representação geral, um vetor final para a palavra alvo.

Segundo Ribeiro, quanto ao *Decoder*, ele recebe a mesma entrada que foi fornecida ao codificador, além de ser alimentado pelo resultado da última camada do *Encoder* em suas subcamadas de *Multi-Head Attention*, assim a segunda subcamada de atenção não é *self-attending*, recebendo as saídas do codificador. Logo, o decodificador possui dois mecanismos de atenção, enquanto o *Encoder* possui apenas um.

Ainda conforme o autor, após passar por todas suas camadas o *Decoder* irá prever uma palavra por vez, baseado na sequência dos *Tokens* (vetor de posições), considerando tanto as palavras já geradas por ele, como pelas representações fornecidas pelo *Encoder*. Desta maneira, vê-se que o decodificador acaba gerando uma palavra após a outra de maneira iterativa.

Pré-Treinamento

O processo de pré-treinamento é crucial na elaboração de um modelo de linguagem, onde, tradicionalmente, este treinamento ocorre de maneira não supervisionada ao considerar duas tarefas: modelagem de linguagem mascarada e previsão da próxima frase (DEVLIN et al., 2019).

Como uma forma de organizar a entrada de texto dois *tokens* especiais "<cls>" e "<sep>", onde um codifica a sentença de entrada e o outro separa as sequências textuais, respectivamente; sendo 512 *Tokens* o comprimento máximo da sequência (ZHANG et al., 2021).

***Masked Language Model* (MLM)** : De acordo com Zhang et al. (2021), a técnica de modelagem de linguagem mascarada – *Masked Language Model* (MLM) – consiste na substituição aleatória de determinados *tokens* por um marcador especial ("<mask>"). Essa estratégia é necessária porque, tradicionalmente, os modelos de linguagem preveem um *token* com base apenas em seu contexto à esquerda,

já conhecido na sequência textual. No caso do BERT, o objetivo passa a ser a predição do *token* mascarado considerando tanto o contexto anterior quanto o posterior, o que viabiliza a aplicação do conceito de bidirecionalidade.

Devlin et al. (2019) descreve o processo do MLM da seguinte forma: o algoritmo seleciona aleatoriamente 15% dos *tokens*, sendo que 80% das vezes o mascaramento é realizado, 10% o *token* selecionado é substituído por outro aleatoriamente, e os outros 10% são inalterados.

Com intuito elucidativo, esse processo é exemplificado abaixo considerando a frase hipotética "João chutou a bola", supondo que "chutou" fora o *token* selecionado de forma randômica:

1. 80% substituído por "<mask>": "João **<mask>** a bola";
2. 10% substituído por *token* aleatório: "João **mesa** a bola";
3. 10% inalterado: "João **chutou** a bola".

Predição da Próxima Sentença : Delvin explicita que a introdução da técnica de predição da próxima sentença – *Next Sentence Prediction* (NSP) – se dá pelo fato de grande parte das demandas de NLP estar fundamentada na compreensão da relação entre duas sentenças, quer perguntas e respostas, quer inferência de linguagem natural. Disso decorre a necessidade de formatar o modelo para que seja capaz de prever a próxima frase a ser gerada.

Seu treinamento, afirma esse autor, se deu da seguinte forma: quando da escolha das frases "a" (primeira) e "b" (sequencia), 50% das vezes "b" é uma sequencia verdadeira e na outra metade é uma frase aleatória do *corpus*.

Exemplificando o supracitado, consideramos a dupla de sentenças: "João chutou a bola. O chute foi forte mas acertou a trave".

1. 50% "b" verdadeiro;
   (a) "João chutou a bola";
   (b) "O chute foi forte mas acertou a trave";
2. 50% "b" substituído por uma frase aleatória;
   (a) "João chutou a bola";
   (b) "O bolo está queimado".

Ajuste Fino

Segundo Devlin et al. (2019), o mecanismo de autoatenção (*self-attention*) dos *Transformers* facilita o processo de ajuste fino (*fine-tuning*) do BERT. Esse processo permite que o modelo seja aplicado em diversas tarefas *downstream* com custo computacional relativamente baixo, já que exige apenas a adição de uma única camada de saída, a qual precisa ser aprendida integralmente pelo modelo.

Essas tarefas podem ser classificadas como segue (ZHANG et al., 2021):

- nível de sequência;
  - **classificação de pares de sentenças**: mede a similaridade entre duas sentenças, como por exemplo:

Sentença "a"	Sentença "b"	Score
"João chutou a bola"	"João chutou a bola"	5.000
"João chutou a bola"	"Maria chutou a bola"	3.000
"João chutou a bola"	"A cadeira é azul"	0.000

  - **classificação de sentenças únicas**: busca classificar uma frase para determinado objetivo, como por exemplo erro gramatical:

Sentença	Classificação
"João chutou a bola"	correto
"João chutou o bola"	incorreto

- nível de *token*;
  - **resposta a perguntas**: o modelo recebe a concatenação entre a pergunta e a frase com possível resposta, e ele precisa encontrar as posições de início e fim das respostas dentro da frase, como por exemplo:

Item	Sentença
Contexto	"Futebol é um dos esportes mais populares no mundo. Mas Pedro gosta de jogar basquete."
Pergunta	"O que Pedro gosta de jogar?"
Resposta	"Basquete."

– **marcação de sentenças únicas**: cada *token* recebe um rótulo, como no caso de identificação de classe gramatical, tal qual na frase "João chutou a bola":

Token	**Classificação**
"João"	substantivo
"chutou"	verbo
"a"	artigo
"bola"	substantivo

Outra aplicação de *fine-tuning*, a extração de frase-chave (*Key Phrase Extraction (KPE)* – principal objeto deste trabalho – pode ser aplicada na resolução de diversas tarefas como recuperação de informações, como resumo, recomendação e recuperação de documentos (SUN et al., 2021).

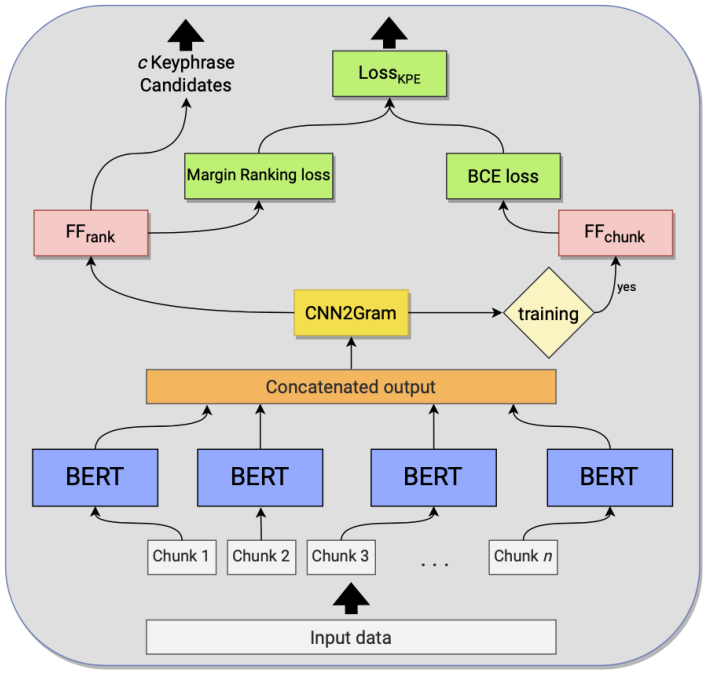
Em seu artigo, Sun et al. (2021) apresenta o modelo "JointKPE", uma técnica supervisionada de KPE que utiliza o BERT para transformar o texto em *embeddings*. A partir dessas representações, n-gramas são construídos por meio de módulos convolucionais.

Em seguida, um classificador linear (*feed forward*) calcula a pontuação de informatividade local de cada n-grama. Posteriormente, n-gramas relacionados são agrupados, de modo a compor a informatividade global, resultando em uma saída com maior poder representativo.

Mesmo com essa capacidade de extrair informação textual, o modelo JointKPE pode apresentar certa limitação ao lidar com textos grandes. Como já mencionado, originalmente o tamanho máximo da sequência de entrada é de apenas 512 *Tokens*, incluindo os especiais "`[CLS]`" e "`[SEP]`". Assim, Alves et al. (2023) propõe um aprimoramento do modelo para que seja capaz de processar até 8.192 *Tokens*, denominado "JointKPE++".

O processo se dá com a divisão do texto em blocos de 512 *Tokens*, e inseri-los no *Encoder BERT*, após o processamento de todos os blocos, eles são concatenados em uma única sequência que alimenta uma camada convolucional, como representado na Figura 2.14, abaixo:

Figura 2.14: Fluxo de processamento JointKPE++



Fonte: Alves et al. (2023).

Hyperparameter Optimization (HPO)

Segundo Lai et al. (2024), a seleção ideal de hiperparâmetros é um fator crítico para maximizar o desempenho e a eficiência de algoritmos de aprendizado de máquina, uma vez que escolhas inadequadas podem levar a processos de aprendizagem subótimos e comprometer a precisão final do modelo.

Esse processo, denominado Otimização de Hiperparâmetros HPO, consiste, conforme Kochnev et al. (2025), em encontrar uma configuração ótima dentro de um espaço de busca que minimize a função de perda (loss function) do sistema.

Para Tribes et al. (2024), no âmbito do Processamento de Linguagem Natural (NLP), a HPO é considerada um *"loop externo"* do aprendizado, onde a automação dessa busca visa superar a natureza ineficiente do ajuste manual realizado a partir da utilização de hiperparâmetros definidos por humanos.

Historicamente, as abordagens tradicionais para HPO dividem-se em busca manual e busca em grade (*grid search*), (LIU; WANG, 2021). Embora a busca em grade seja sistemática, ela sofre com a baixa eficiência devido ao aumento

exponencial do custo computacional em relação ao número de hiperparâmetros analisados.

Para mitigar essas limitações, Lai et al. (2024) discorre que métodos automatizados mais avançados têm sido desenvolvidos, incluindo a busca aleatória (*random search*), estratégias evolutivas e a otimização bayesiana.

A otimização bayesiana, especificamente, destaca-se por construir um modelo substituto (surrogate model) que é refinado iterativamente para guiar a busca em direção a regiões do espaço de parâmetros com maior probabilidade de sucesso, de acordo com Kochnev et al. (2025).

Mulakala et al. (2024) comenta que, no contexto específico do ajuste fino (*fine-tuning*) de modelos de linguagem pré-treinados, a HPO torna-se um desafio acentuado devido ao grande volume de parâmetros e ao elevado custo de processamento. Hiperparâmetros como a taxa de aprendizado (*learning rate*), o tamanho do lote (*batch size*), o número de épocas e o *dropout* são fundamentais para garantir que o modelo generalize corretamente para novas tarefas.

BERTimbau

Baseado no conceito de transferência de conhecimento (*transferring learning*), o modelo BERT faz uso de uma base pré-treinada, o que possibilita reduzir os custos de processamento e viabilizar outros projetos sem a necessidade de iniciar o treinamento a partir do zero.

Muitas bases disponíveis que podem ser utilizadas no BERT são pré-treinadas para o idioma inglês. No entanto, como o presente projeto tem o enfoque em lidar com textos de mensagens de celulares capturados no Brasil, optou-se pelo uso de uma base treinada em língua portuguesa, no caso o BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020).

Segundo Souza (2020), esse modelo foi treinado com a base *brWaC* (Brasil *web as Corpus*), sendo um grande *corpus* de textos da *web*, composto por 2,68 bilhões de *Tokens* de 3,53 milhões de documentos. Após uma etapa de *data-cleaning*, o *corpus* continha 17,5 GB de texto puro.

O modelo foi treinado em dois tamanhos:

- ***Base:*** 12 camadas, 768 dimensões ocultas, 12 cabeças de atenção (*Multi-head attention*) e 110 milhões de parâmetros; e
- ***Large:*** 24 camadas, 1024 dimensões ocultas, 16 cabeças de atenção (*Multi-head attention*) e 330 milhões de parâmetros;

O BERTimbau obteve melhores resultados que o estado da arte de modelos BERT multilíngue e anteriores abordagens monolíngues, tendo sido avaliado mas tarefas de similaridade textual (*Sentence Textual Similarity (STS)*), reconhecimento de inferência textual (*Recognizing Textual Entailment (RTE)*) e reconhecimento de entidade nomeada (*Named Entity Recognition (NER)*).

Na Tabela 2.2, os resultado obtidos pelo BERTimbau, destaca-se os valores obtidos com RTE, visto que se trata da técnica aplicada nesta pesquisa:

Tabela 2.2: Comparação dos resultados RTE obtidos pelo BERTimbau em relação com outros modelos.

		RTE	
Row	**Model**	**F1 (*)**	**Accuracy**
1	mBERT + RoBERTa-Large-en (Averaging) (RODRIGUES et al., 2020b)	84	84.8
2	mBERT + RoBERTa-Large-en (Stacking) (RODRIGUES et al., 2020b)	88.3	88.3
3	mBERT (STS) and mBERT-PT (RTE) (RODRIGUES et al., 2020a)	87.6	87.6
4	USE+Features (STS) and mBERT+Features (RTE) (FONSECA; ALVARENGA, 2020)	86.6	86.6
5	mBERT+Features (FONSECA; ALVARENGA, 2020)	86.6	86.6
6	mBERT (ours)	86.8	86.8
7	BERTimbau Base	89.2	89.2
8	BERTimbau Large	90.0	90.0

Fonte: Elaborado pelo autor, adaptado de Souza (2020).

Grandes Modelos de Linguagem (LLM)

Tido como os modelos mais modernos na atualidade, os Grandes Modelos de Linguagem (LLM) promoveram um grande salto tecnológico, inclusive extrapolando o ambiente acadêmico e se democratizando à sociedade, em meados dos anos 2022 e 2023 como a disponibilização desta inteligência ao grande público através de modelos de *chat* onde pessoas ao redor do mundo puderam interagir com esses modelos via internet.

Mas, antes de aprofundar neste tema, importa que seja esclarecido que os LLM são modelos também pré-treinados baseados em *Transformers* contendo centenas de bilhões de parâmetros, treinados com grande massa de textos, con-

forme Zhao et al. (2023), o que acabou lhes proporcionando um enorme poder de resolver problemas complexos por meio da geração de texto.

Convém destacar que sua construção ser baseada na arquitetura *Transformes* é semelhante aos modelos *Pre-trained Language Model* (PLM), mas que devido sua imensa base de treinamento e computação ampliou seu desempenho.

Segundo Zhao et al. (2023), alguns pontos importantes diferenciam os modelos grandes (LLM) dos pequenos (PLM), as habilidades emergentes. Estas se caracterizam pelo surgimento de uma capacidade dos grandes modelos em realizar uma tarefa complexa que os modelos pequenos (PLM) não eram capazes de resolver, como:

- Aprendizagem contextual: *in-context learning in-context learning* (ICL) onde a partir de uma contexto de entrada, instruções de linguagem natural ou demonstrações de tarefas, o modelo é capaz de gerar o resultado esperado a partir do contexto fornecido;
- Seguimento de instrução: dada a aplicação da técnica de *fine-tuning* os LLM são capazes de melhorar sua capacidade de generalização, quando conseguem seguir instruções de tarefas dadas e resolver novas tarefas sem a necessidade de novos exemplos explícitos; e
- Raciocínio passo-a-passo: capacidade de resolução de problemas por etapas, comum em questões matemáticas com descrições textuais, para tal esses modelos empregam a estratégia de cadeia de pensamento, *chain-of-thought (CoT)*.

Ainda de acordo com o autor, algumas técnicas foram cruciais para que esse modelos evoluíssem até o estado atual, sendo elas:

- Dimensionamento: um efeito direto onde o tamanho do modelo, da base de dados e da capacidade de treinamento, tende a melhora a capacidade de resolução de problemas;
- Treinamento: devido a complexidade computacional exigida para o treinamento desses modelos com grande capacidade, faz-se necessário o emprego de técnicas de processamento distribuído e paralelismo;
- Capacidade de elicitação: após treinamento massivo com a *corpora* os LLM são dotados de capacidade de solucionar problemas de propósito geral, ou

seja, ser capazes de gerar uma resposta a partir de instruções em linguagem natural, ou por aprendizagem a partir de um contexto;

- Ajuste de alinhamento: pelo fato desses modelos serem treinados a partir de uma grande base de dados, podendo ser composta por itens de boa e também de má "qualidade" (conteúdo tóxico, tendencioso ou prejudicial às pessoas), faz-se necessário que eles sejam ajustados a fim de se alinhar ao valores humanos como honestidade, utilidade e inocuidade; e
- Manipulação de ferramentas: como são essencialmente geradores de texto formados a partir de uma enorme *corpora* de textos simples, falta-lhe a habilidade de lidar com tarefas não expressas em texto, como no caso de cálculo numérico, bem como fornecer respostas precisar a cerca de informações atuais; assim, uma das formas de resolver esse problema foi a utilização de ferramentas externas, como no caso do emprego de calculadoras de cálculos precisos, ou mecanismos de buscas a fim de encontrar respostais mais atuais à sua base de treinamento.

Convém ressaltar que esses modelos possuem uma importante variável de ajuste do nível de determinismo de suas saídas, a **temperatura**. Segundo, Holtzman et al. (2020), este é um hiperparâmetro que ajusta a suavidade da distribuição de probabilidades produzida pela função softmax. Valores mais baixos reduzem a entropia da distribuição, privilegiando escolhas mais prováveis, ao passo que valores mais altos aumentam a diversidade.

Outro ponto de destaque nessa jornada evolutiva foi o alinhamento desses modelos às preferências humanas através da técnica de aprendizagem por reforço com *feedback* humano (*reinforcement learning from human feedback* (RLHF)), a qual consiste num processo composto de três etapas: a) ajuste fino supervisionado (*supervised fine-tuning*), no intuito de se obter comportamentos do LLM previamente definidos é necessário que este seja alimentado com uma base de dados supervisionados com instruções e saídas desejadas, muitas vezes tal anotação escrita por humanos; b) treinamento de modelo de recompensa (*reward model training*), após a geração de amostra de saídas, rotuladores (pessoas responsáveis por rotular os dados) anotam suas preferências no intuito de alinhar o modelo às preferências humanas; c) ajuste fino de aprendizagem por aprendizagem (*reinforcement learning fine-tuning*), nesta etapa o ele é ajustado a partir de técnicas de recompensas e penalidades em relação às divergências entre os resultados gerados pelo modelo ajustado e o inicial.

Mas como esses modelos alcançaram essa capacidade de generalização? Os LLM foram treinados com uma gigantesca base de dados (ZHAO et al., 2023), uma *corpora* composta por páginas da internet, textos de conversas, livros, textos de vários idiomas (não se limitando ao inglês), textos científicos e códigos de programação. Obviamente que cada modelo com uma base adaptada ao seu objetivo como escrita de códigos de programação, compreensão de linguagem coloquial, ou formal.

Mesmo com tamanha capacidade de "compreensão" linguística, ele pode apresentar uma limitação em relação a conteúdo especializado, como termos técnicos de determinada área do conhecimento, havendo a necessidade de incorporar esse conhecimento ao modelo. Todavia, pode ocorrer que uma vez ajustado à determinada temática ele possa encontrar dificuldades em lidar com outras, isso podendo ligado ao esquecimento catastrófico (*catastrophic forgetting*) – um conflito de integração entre conhecimentos novos e antigos – devendo levar em consideração essa "taxa de alinhamento", relativa ao ganho de ajuste às necessidades humanas em detrimento de seu generalismo.

Não obstante esse patamar epistemológico alcançado, os LLM podem ser acometidos, principalmente, por problemas como alucinação, atualização de conhecimento e resolução de tarefas de raciocínio complexo (ZHAO et al., 2023).

- Alucinação: o modelo pode gerar uma resposta que não condiga com a realidade, com sua base de treinamento podendo resultar em uma informação "inventada" pelo modelo, ou mesmo uma resposta incorreta ou imprecisa;
  - intrínseca: quando a resposta é conflita diretamente com a fonte existente, como no caso do fornecimento da data errada de um evento histórico;
  - extrínseca: que não podem ser verificadas pela fonte disponível, como a inserção de uma informação incorreta dentro de uma resposta com conteúdo correto, improvisando informações alheias à fonte;
- Atualização de conhecimento: os modelos podem não possuir informações atualizadas, novos conhecimentos gerados pela sociedade, isso pelo fato de que seu ajuste, geralmente, dispende de altos valores, além do risco de ocorrência do esquecimento catastrófico, acima explicitado. Como possível solução, ainda que não proporcione uma solução definitiva, pesquisas

recentes apresentam o uso em conjunto de mecanismos de busca para exploração de conhecimento externo; e

- Resolução de tarefas de raciocínio complexo: o LLM pode apresentar certa inconsistência especialmente resultante dum processo de raciocínio decomposto, podendo gerar uma resposta correta a partir de uma construção equivocada, ou mesmo gerar uma resposta incorreta, além do fato de que pequenas mudanças na entrada podem produzir resultados distintos; também com relação ao cálculo numérico, visto que alguns símbolos podem não ser tão comuns em sua base de treinamento.

Ainda que tendo de lidar com tais dificuldades, os Grandes Modelos de Linguagem (LLM) ainda são um vasto campo a ser explorado e aperfeiçoado com possibilidade de fornecer grande auxílio em demandas da sociedade.

2.6 Modelos Transcritores

Dado o objeto de estudo desta pesquisa, mensagens trocadas em *chats* de dispositivos móveis, tem-se que, atualmente, tais aplicativos de conversa operam com mais de um modo de comunicação, como: escrita, áudio, vídeo ou imagens.

Assim, neste estudo optou-se por trabalhar com mensagens textuais e de áudio. Logo, para que essas mensagens de voz fossem passíveis de análise textual, foi necessário utilizar técnicas de transcrição de áudio.

Desta maneira, uma vez identificada a existência de um áudio durante a troca de mensagens, esse fora transcrito e inserido na sua respectiva posição de emissão.

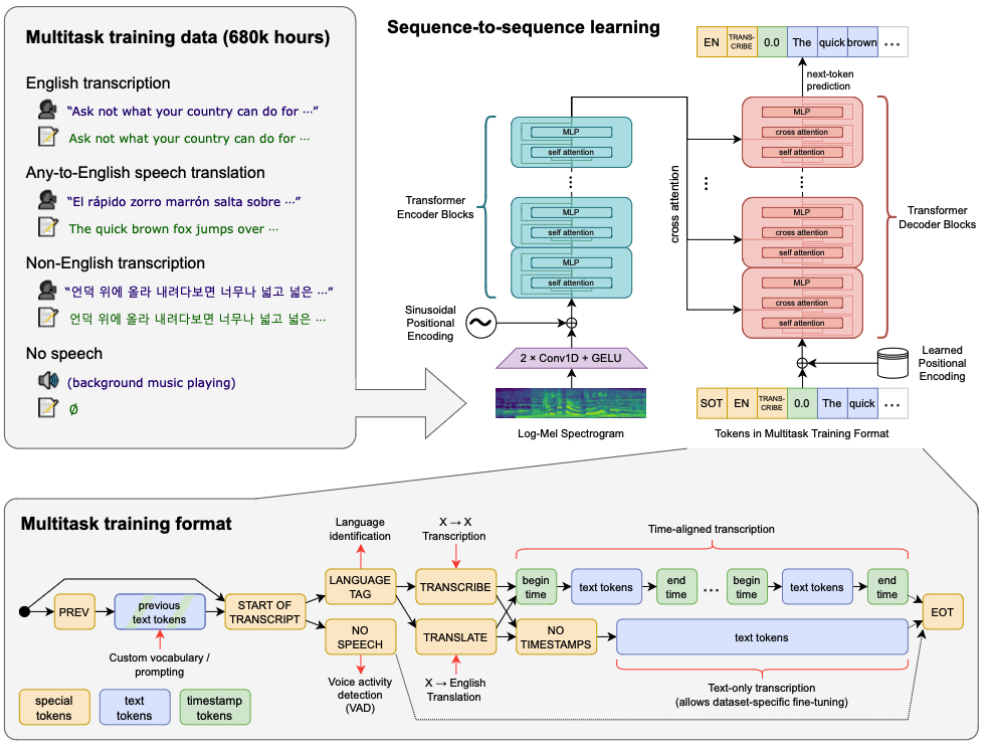
Para execução desta atividade, foram selecionado para testes dois modelos passíveis de execução *on premise*: Whisper e Vosk, os quais são detalhados nesta seção.

Para a execução desta atividade, foram selecionados dois modelos passíveis de execução *on premise*: Whisper e Vosk, os quais são detalhados nesta seção. A escolha por soluções locais justifica-se pela natureza sensível dos dados forenses, que impede sua transmissão a serviços externos via *internet*, garantindo assim maior segurança, privacidade e conformidade com protocolos periciais.

2.6.1 Whisper

O Whisper, desenvolvido pela OpenAI (RADFORD et al., 2023), é um modelo de transcrição baseado na arquitetura *Transformers* (apresentada na Subseção 2.5.4). Ele é capaz de realizar reconhecimento de fala multilíngue, tradução de áudio, identificação do idioma falado e detecção de atividade de voz. Sua abordagem para execução da transcrição está detalhada na Figura 2.15.

Figura 2.15: Abordagem de transcrição de áudio - Whisper

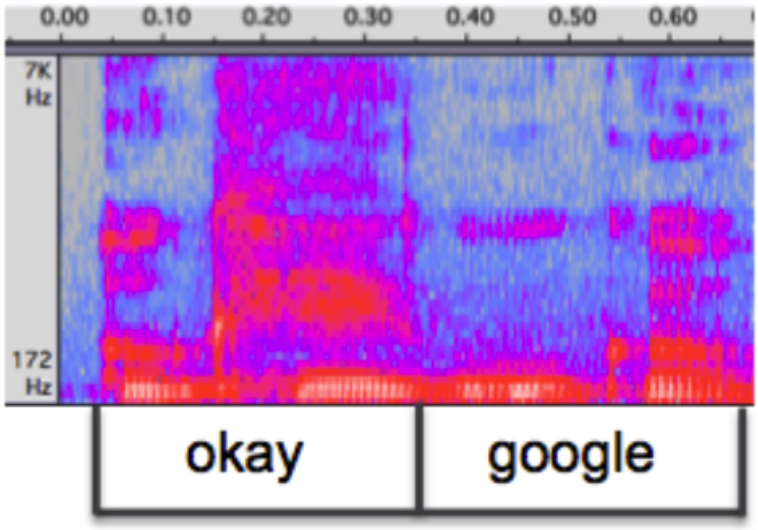


Fonte: Radford et al. (2023).

Primeiramente, o modelo converte o áudio em um espectrograma Log-Mel (RADFORD et al., 2023). Um espectrograma, conforme Lacerda et al. (2021), é uma representação visual da variação de um sinal ao longo do tempo. Nele, as frequências são dispostas no eixo vertical e o tempo no eixo horizontal, enquanto a intensidade da energia em cada instante é indicada por variações de cor.

A Figura 2.16 ilustra esse conceito por meio da representação da frase "okay google", mostrando a evolução de suas frequências ao longo do tempo.

Figura 2.16: Abordagem de transcrição de áudio - Whisper



Fonte: Chen (2014, p. 1) *apud* Lacerda (2021, p. 3).

Como mencionado anteriormente, o Whisper converte os sinais em um *Mel-spectrogram*, que corresponde a um espectrograma tradicional (Figura 2.16), porém transformado para a escala Mel.

A conversão para essa escala é definida pela Equação 2.15, conforme apresentado em Lacerda et al. (2021):

$$\text{Mel}(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.15}$$

Essa transformação aplica uma escala logarítmica com o intuito de enfatizar as frequências mais perceptíveis ao ouvido humano. O objetivo central é reduzir a quantidade de informação representada, preservando apenas os aspectos acústicos mais relevantes, sem perdas significativas de conteúdo (LACERDA et al., 2021).

Assim, de acordo com Radford et al. (2023), o transcritor converte todos os áudios em representações de espectrograma Mel de 80 canais, sendo processado por um *encoder* numa camada inicial composta por duas camadas convolucionais e uma função de ativação GELU.

Sequencialmente os *embedding* são adicionados à saída da camada inicial, sendo aplicados os blocos de *encoder* do *Transformer*, permitindo seu treinamento

num formato multitarefa.

O autor conclui a descrição da *pipeline* informando que, ao final, o *decoder* utiliza os *embeddings* de posição aprendidos para prever os *tokens* de saída.

O modelo Whisper possui cinco tamanhos (Tiny, Base, Small, Medium e Large), que se diferenciam por hiperparâmetros arquiteturais como o número de camadas, a largura dos blocos, o número de cabeças de atenção e o total de parâmetros (Tabela 2.3).

Tabela 2.3: Modelos Whisper de diferentes tamanhos

Model	**Layers**	**Width**	**Heads**	**Parameters**
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Fonte: Radford et al. (2023)

Em termos práticos, modelos maiores tendem a gerar embeddings mais expressivos e, consequentemente, oferecer melhor qualidade de transcrição. Eles se destacam em cenários complexos, como áudios com ruído, sotaques ou enunciados longos. No entanto, essa maior capacidade vem com um custo computacional elevado: demandam mais memória, tempo de processamento, energia e uma dependência mais rígida de GPU.

No contexto deste trabalho, que exige a execução on premise sobre um dataset volumoso, a escolha do modelo ideal é um equilíbrio entre qualidade e custo computacional. Modelos menores são mais rápidos e leves, mas menos precisos, enquanto modelos maiores garantem ganhos de qualidade, mas com maior demanda computacional e tempo de execução. Portanto, a seleção do modelo deve considerar não apenas o desempenho esperado, mas também a viabilidade de processamento local dos áudios em um ambiente forense.

2.6.2 Vosk

Segundo Martins et al. (2021), trata-se de uma plataforma de reconhecimento de fala com código aberto, com capacidade de lidar com 17 idiomas, como modelos de 50 MB, porém permitem a escolha de modelos maiores, ficando à critério do usuário.

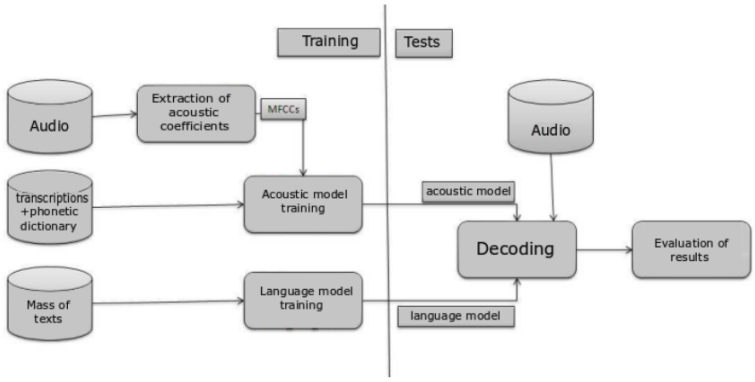
Especificamente, para a língua portuguesa, o Vosk[1] possui dois modelos: *small* com 31M e o *large* de 1.6G.

Ainda de acordo com Martins et al. (2021), a plataforma possui integração com o Kaldi (POVEY et al., 2011), ferramenta de reconhecimento de fala em código aberto que utiliza redes neurais profundas (*Deep Neural Networks* (DNN)). Essas redes estão implementadas em três *codebases* distintas (diferentes implementações ou gerações), a saber:

- **nnet1**: mais simples, com treinamento em apenas uma única GPU;
- **nnet2**: permite uso de múltiplas GPUs ou CPUs;
- **nnet3**: projetada para suportar redes neurais mais gerais e avançadas.

A transcrição textual, consiste num processo de conversão de um áudio em texto, onde primeiramente é realizado o treinamento do modelo extraindo coeficientes acústicos dos áudios que serão utilizados, posteriormente, como referência na fase de decodificação. Na fase de teste, ou aplicação, a decodificação do áudio de entrada é realizada com emprego de dois modelos pré-treinados: de áudio e de linguagem, conforme Figura 2.17.

Figura 2.17: Processo de reconhecimento de fala



Fonte: Martins et al. (2021).

[1]VOSK. Vosk recognition. Disponível em: <https://alphacephei.com/vosk/>. Acesso em: 13 jun. 2024 às 13:17h.

2.7 Considerações Finais

Neste segundo capítulo foi apresentada a base conceitual para a pesquisa, formalizando termos técnicos de NLP como *tokens*, sentenças e *embeddings*, fundamentais para o processamento textual.

Também foram detalhados os modelos de extração de frases-chave (como TF-IDF, RAKE e TextRank), os principais classificadores de Machine Learning do tipo supervisionado (incluindo SVM, *Naive Bayes*, MLP, *Random Forest* e XGBoost), e os modelos de linguagem baseados em redes neurais, com ênfase na arquitetura *Transformer*, no BERT (e BERTimbau) e nos Grandes Modelos de Linguagem (LLM). Além disso, foram apresentados os modelos transcritores (Whisper e Vosk) utilizados na conversão de mensagens de áudio em texto.

Com o entendimento consolidado das técnicas e modelos, na sequência deste trabalho serão contextualizadas as aplicações dessas ferramentas, revisando os Trabalhos Relacionados que exploram a identificação de conteúdo ilícito e a análise forense em ambientes digitais.

3

TRABALHOS RELACIONADOS

A aplicação de técnicas de Processamento de Linguagem Natural (NLP) juntamente com arquiteturas de *Machine Learning* (ML) veêm ganhando destaque dentro da academia, ainda mais nos últimos anos, devido o grande desenvolvimento da computação, especialmente no uso da inteligência artificial na automação de atividades humanas.

Estudos focados na análise textual com uso de processamento de linguagem, têm dominado temas da área de informática. Isso devido a crescente quantidade de dados gerados, bem como o aumento do poder de processamento, o que permite uma avaliação mais abrangente, rápida e automatizada do vasto conteúdo digital gerado pela sociedade.

Com isso, vê-se o aparecimento de diversas pesquisas ligadas à análise de textos, especialmente voltadas às mídias sociais como: Whatsapp, Instagram, Facebook, entre outros. Tal enfoque acaba se justificando pelo fato de ser uma grande fonte de dados, atualizados e que representam exposições autênticas do cotidiano e opiniões da sociedade, justificando o crescente interesse acadêmico e científico.

Esta capítulo apresenta trabalhos que realizaram aplicação de NLP na análise de textos, quer relacionados à classificação de sentenças, rotulação de dados ou análise forense voltadas à identificação de atividades criminosas.

3.1 Identificação de Conteúdo Ilícito em Redes Sociais

Diversos estudos têm explorado o uso de modelos de linguagem e técnicas de ML na identificação automatizada de conteúdo ilícito, especialmente em plataformas como o Instagram. A seguir, são apresentados trabalhos que aplicam modelos de NLP na tarefa de detecção de postagens associadas ao tráfico de drogas.

As redes sociais tem sido cada vez mais usadas como veículos de disseminação de venda de drogas, criando novos desafios na identificação de atividades e comportamentos suspeitos, ainda mais com as recentes legalizações de certas drogas, sendo necessário um maior refinamento na classificação. A pesquisa realizada por Hu et al. (2023b), intitulada como "Fine-grained classification of drug trafficking based on Instagram hashtags", tem por base a identificação e classificação de *hashtags* relacionadas ao tráfico de drogas nas postagens ou comentários do Instagram.

Sua base de dados formada por 20.000 postagens e comentários, sendo que desses 34.5% foram anotados manualmente. O uso do BERT se deu para a extrair o contexto semântico das *hashtags* e as relações entre eles com a aplicação da técnica de grafos. A valiação de desempenho foi feita com o uso de validação cruzada de 10 *folds*, tendo alcançado um resultado de *F1-Score* de 75%.

Outra pesquisa voltada para a identificação de conteúdo relacionado ao tráfico de drogas em textos postados na rede social Instagram é apresentada no trabalho "Unveiling the Potential of Knowledge-Prompted ChatGPT for Enhancing Drug Trafficking Detection on Social Media" (HU et al., 2023a). O estudo destaca a utilização de modelos de Grandes Modelos de Linguagem (LLM) na execução dessa tarefa, com ênfase na aplicação de engenharia de *prompts*.

A base de dados utilizada consistiu em 886 amostras de textos, sendo 486 exemplos positivos (relacionados ao tráfico de drogas) e 400 negativos. O processamento foi realizado utilizando a API do modelo ChatGPT 3.5 Turbo, onde os textos foram analisados juntamente com *prompts* informados por conhecimento, projetados para orientar o modelo na classificação. Como resultado, o estudo alcançou um *F1-Score* de 94,98%, demonstrando a eficácia da abordagem baseada em *prompts* na adaptação do LLM para esta tarefa específica.

Para além da análise textual, o artigo "Identifying Illicit Drug Dealers on

Instagram with Large-scale Multimodal Data Fusion" de Hu et al. (2021) aborda a integração de diversos tipos dados da rede social Instagram, incluindo comentários em postagens, imagens publicadas, biografias de usuários e imagens das páginas iniciais dos perfis.

Segundo a pesquisa, das cerca de 4.000 contas avaliadas, mais de 1.400 foram identificadas como pertencentes a traficantes de drogas (positivos), enquanto cerca de 2.000 foram classificadas como não relacionadas (negativos). Tal constatação se deu a partir da rotulação dos dados (textos e imagens), que devido sua complexidade e volume, demandou aproximadamente 400 horas de execução contando com 10 participantes.

Para a análise, o modelo BERT foi empregado na classificação de dados textuais, enquanto o modelo ResNet-50 foi utilizado para imagens, ambos com aplicação da técnica de *fine-tuning*. As representações vetoriais (*embeddings*) geradas por cada modelo foram concatenadas, combinando as 768 dimensões do BERT com as 2048 dimensões do ResNet-50. Essa abordagem multimodal atingiu um desempenho de F_1-Score superior a 94% quando considerada a concatenação das características extraídas das quatro modalidades avaliadas.

Ainda, o estudo "Tracking Illicit Drug Dealing and Abuse on Instagram Using Multimodal Analysis" (YANG; LUO, 2016), propõe uma metodologia que combina aprendizado multi-tarefa e fusão de decisões em diferentes níveis, propondo um *framework* dividido em duas etapas: análise em nível de postagem para reconhecer posts relacionados a drogas com alta precisão, e análise em nível de conta para identificar padrões de comportamento e detectar contas de traficantes. Os resultados finais demonstraram uma capacidade de detectar posts relacionados a drogas com alta precisão (cerca de 88%) e identificar contas de traficantes com um *F1-Score* de aproximadamente 0,51. Os resultados indicam que técnicas automatizadas de análise multimodal podem ser eficazes para combater o tráfico de drogas nas redes sociais, especialmente no Instagram, onde os traficantes utilizam conteúdo visual e *hashtags* para promoção e venda ilegal.

3.2 Análise de Interações e Comportamentos Ilícitos em Plataformas Digitais

Para além da análise de conteúdo, há também um corpo relevante de pesquisas voltado à modelagem das interações entre usuários e à dinâmica dos mercados ilegais nas redes sociais. Tais abordagens incluem desde métodos de aprendizado não supervisionado até investigações qualitativas baseadas em observação e entrevistas.

Nessa linha, o artigo "An unsupervised machine learning approach for the detection and characterization of illicit drug-dealing comments and interactions on Instagram" de Shah, Li & Mackey (2022) propõe uma abordagem não supervisionada para identificação de interações de tráfico nos comentários do Instagram, com uso do modelo Biterm Topic Model (BTM) para clusterização temática. A partir de mais de 43 mil comentários coletados por meio de *hashtags* relacionadas a drogas, os autores identificaram 5.589 comentários com indícios de comércio ilegal, predominantemente oriundos de vendedores e farmácias online (99,7%). Os principais temas estavam relacionados à menção a preços, disponibilidade e dados de contato via aplicativos criptografados. A metodologia não supervisionada mostrou-se promissora especialmente para cenários com escassez de dados rotulados.

No artigo "Drug dealing on Facebook, Snapchat and Instagram: A qualitative analysis of novel drug markets in the social media environment", Demant et al. (2019) investigou o comportamento de traficantes e compradores de drogas em plataformas sociais populares por meio de entrevistas – 127 participantes com idade entre 16 e 45 anos – e observações online. A pesquisa revelou alto grau de atividade relacionada ao tráfico de drogas no Facebook, Instagram, Snapchat e Facebook Messenger, onde tais mídias acabam se tornando ferramentas para na comercialização de ilícitos, onde muitos dos participantes relatam que entram e saem facilmente das redes sociais para negociar e comprar, sem estarem cientes da gravidade do delito.

3.3 Análise Forense e Processamento de Mensagens em Ambientes Criptografados

O uso de aplicativos de mensagens criptografadas como o WhatsApp impõe desafios adicionais à investigação de práticas ilícitas. Neste contexto, estudos têm buscado desenvolver soluções forenses e modelos de NLP para extrair informações úteis a partir de textos compartilhados em grupos.

Como exemplo, a pesquisa "Identifying interception possibilities for WhatsApp communication" (WIJNBERG; LE-KHAC, 2021) preconiza os desafios enfrentados por autoridades policiais diante do uso crescente de aplicativos de mensagens criptografadas como o *WhatsApp*. O artigo propõe uma estrutura forense para obtenção de informações em tempo real a partir de múltiplas fontes — interceptações, inteligência de código aberto, análise do *WhatsApp Web* e descriptografia de backups. Dentre os dados acessíveis pela abordagem proposta estão localizações, imagens, documentos e áudios compartilhados, ampliando o escopo investigativo para além da interceptação de metadados.

O trabalho "Enhancing Text Sentiment Classification with Hybrid CNN-BiLSTM Model on WhatsApp Group" (SUSANDRI; DEFIT; TAJUDDIN, 2024) explora o uso de aprendizado profundo para análise de sentimento em mensagens trocadas em grupos. O modelo proposto combina redes convolucionais (CNN) com redes recorrentes bidirecionais de memória longa (BiLSTM). Os resultados demonstram que, em testes de modelo único, a LSTM e a BiLSTM alcançaram uma precisão de 81%. O modelo híbrido proposto, no entanto, atingiu uma precisão de 88% no conjunto de dados utilizado. Em comparação com estudos anteriores, o modelo híbrido superou o desempenho, indicando uma melhor capacidade de classificação de sentimentos.

Dessa forma, evidencia-se uma convergência entre abordagens qualitativas, técnicas de aprendizado multimodal e métodos forenses digitais como estratégias complementares na identificação de práticas ilícitas em redes sociais e aplicativos de mensagem. Essa diversidade metodológica revela-se essencial frente à constante adaptação dos atores criminosos ao ambiente digital, exigindo soluções igualmente dinâmicas e integradas por parte da comunidade científica.

3.4 Modelagem e Classificação de Textos Jurídicos e Criminais em Língua Portuguesa

O trabalho de Carnaz, Antunes & Nogueira (2021) apresenta a criação do primeiro corpus anotado dedicado ao domínio criminal na língua portuguesa, visando suprir a carência de recursos para o processamento de relatórios policiais. Os autores coletaram e anonimizaram documentos de fontes oficiais, como relatórios de investigação e notícias criminais, mantendo o contexto original para permitir a extração automática de correlações entre suspeitos, locais e eventos. Esse conjunto de dados foi estruturado em formato XML para facilitar o treinamento de modelos de aprendizagem de máquina voltados à segurança pública.

A metodologia empregada envolveu a anotação manual de entidades nomeadas e um conjunto anotado para extração de informações. Além de categorias tradicionais como pessoas e organizações, a pesquisa incluiu marcadores específicos para tipos de crimes e narcóticos, abrangendo tanto a terminologia oficial quanto gírias e expressões coloquiais do ambiente criminal.

Os resultados experimentais indicaram um *F1-Score* médio de 0,73 na tarefa de Reconhecimento de Entidades Nomeadas (NER) e de 0,65 na extração de informações baseada em 5W1H. Para o domínio específico de entorpecentes, o modelo alcançou um *F1-Score* de 0,77 utilizando validação cruzada, o que evidencia a eficácia da abordagem na identificação de termos relacionados ao tráfico. O estudo conclui que a disponibilidade de bases rotuladas é o fator determinante para o sucesso da aplicação de NLP em investigações forenses.

A pesquisa de Raulino et al. (2021) investiga o uso de inteligência artificial para analisar o crescente volume de processos no Judiciário brasileiro, focando especificamente em decisões do Supremo Tribunal Federal (STF) sobre prisão preventiva. O estudo coletou um conjunto de dados de 2.200 julgados para prever, por meio de classificação de texto, se o tribunal concederia ou não a liberdade a um prisioneiro provisório em sede de habeas corpus.

Tecnicamente, o estudo comparou algoritmos clássicos com redes neurais profundas, utilizando técnicas de pré-processamento como stemming e análise de N-Grams em português. As representações textuais foram baseadas em Word Embeddings (GloVe) treinadas especificamente com textos do domínio jurídico brasileiro. Entre os modelos testados, a Rede Neural Convolucional

(CNN) apresentou o desempenho superior, atingindo uma acurácia de 95% e umc*F1-Score* de 0,91 na previsão dos resultados dos julgamentos.

Além da classificação, os autores aplicaram o algoritmo FP-Growth para identificar regras de associação entre o tipo de crime e o desfecho processual. Os resultados revelaram uma forte correlação estatística entre crimes da lei de drogas e a manutenção da prisão preventiva (resultado "não liberado"). O trabalho ressalta que tais modelos não visam substituir a avaliação humana, mas servir como instrumentos de triagem e redução de vieses em decisões que afetam a liberdade individual.

O estudo de Santos et al. (2025) realiza uma comparação sistemática entre três paradigmas tecnológicos para a classificação de petições jurídicas em português: modelos clássicos, modelos *Transformer* com ajuste fino *(fine-tuning)* e Grandes Modelos de Linguagem (LLM) baseados em *prompting*. A pesquisa introduziu um novo conjunto de dados composto por 3.458 documentos de uma Defensoria Pública (DPE-GO), rotulados em 24 categorias procedimentais. O objetivo central foi avaliar o equilíbrio entre eficácia preditiva, custo computacional e necessidade de dados rotulados.

Os experimentos demonstraram que o *BERTimbau Large*, após o processo de *fine-tuning*, obteve o melhor desempenho absoluto, alcançando um *F1-Score* de 94,70%. Esse resultado confirma que modelos monolíngues pré-treinados no idioma local capturam melhor as nuances da linguagem jurídica do que arquiteturas multilíngues genéricas. Por outro lado, modelos clássicos como KNN mostraram-se competitivos em cenários de recursos limitados, mantendo boa interpretabilidade.

No âmbito dos LLM, a estratégia baseada em sumários apresentou os melhores resultados (*F1-Score* de 88,52%). O estudo destaca que a redução do texto para versões condensadas minimiza o ruído informacional e otimiza o consumo de *tokens*. A pesquisa conclui que a escolha da técnica deve depender da disponibilidade de infraestrutura, tendo obtido o maior desempenho a partir da aplicação do modelo pré-treinado juntamente com a técnica de *fine-tuning*.

3.5 Adaptação de Domínio e Evolução de Modelos Pré-Treinados

O trabalho de Silva et al. (2024) avalia a eficácia da técnica de *Domain-adaptive pre-training* (DAPT) para especializar modelos de linguagem no domínio governamental brasileiro. A premissa da pesquisa é que modelos treinados em corpora genéricos apresentam dificuldades para compreender o vocabulário especializado e a sintaxe complexa de documentos administrativos e licitatórios. Para isso, os autores realizaram o pré-treinamento contínuo dos modelos BERTimbau e LaBSE utilizando grandes volumes de dados não rotulados de diários oficiais e tribunais superiores.

A metodologia variou sistematicamente a composição linguística do modelo (monolíngue vs. multilíngue) e o tamanho da base de dados de pré-treinamento para medir o impacto na classificação de itens e documentos. Os modelos foram submetidos à tarefa de *Masked Language Model* (MLM), onde aprenderam a prever *tokens* ocultos dentro do contexto técnico governamental. Os resultados mostraram que o uso de dados de domínios correlatos (como textos jurídicos para classificar licitações) melhora significativamente a capacidade de generalização do modelo.

Uma das principais conclusões foi que modelos monolíngues baseados no BERTimbau superaram consistentemente as versões multilíngues em tarefas de alta especificidade, devido ao entendimento refinado das nuances do português brasileiro. O estudo também observou que o aumento massivo da base de dados nem sempre resulta em ganhos lineares de performance, sugerindo que a qualidade e a relevância do corpus são mais cruciais do que apenas o seu volume. Essa abordagem de adaptação é diretamente relevante para o ajuste de classificadores periciais em contextos de baixa disponibilidade de dados rotulados.

Scalercio et al. (2025) aborda em seu artigo a tarefa de simplificação de sentenças em português, visando tornar textos técnicos de áreas como direito e administração pública mais acessíveis ao cidadão comum. O estudo introduziu o *corpus Gov-Lang-BR*, composto por 1.703 pares de sentenças complexas e suas versões simplificadas provenientes de agências governamentais brasileiras. Esse trabalho é inovador ao realizar um *benchmark* abrangente com 26 LLM diferentes para avaliar a capacidade de transformação linguística desses modelos.

Os autores utilizaram uma estratégia de *one-shot prompting*, fornecendo exemplos de simplificações sintáticas e lexicais para guiar a geração do modelo. Embora modelos proprietários de larga escala (GPT-4o-mini) tenham liderado o *ranking*, o estudo demonstrou que modelos de código aberto (*open-weight*) como a família Qwen-2.5 apresentam resultados competitivos e podem ser executados localmente por meio de técnicas de quantização para 4-bit.

A análise linguística detalhada revelou que, apesar do avanço tecnológico, ainda existe uma lacuna entre as simplificações produzidas por máquinas e por humanos, especialmente na conversão de voz passiva para ativa e na redução de orações relativas. O estudo destaca que a escolha do exemplo fornecido no *prompt* (*one-shot*) influencia drasticamente a qualidade final, sendo os exemplos de edição sintática os mais eficazes para gerar textos claros.

3.6 Considerações Finais

O Capítulo 3 revisou a literatura pertinente, demonstrando como as técnicas de Processamento de Linguagem Natural (NLP) e *Machine Learning* (ML) têm sido empregadas na identificação automatizada de conteúdo ilícito e na análise forense.

Foram abordados estudos focados na detecção de tráfico de drogas em redes sociais como o *Instagram*, utilizando abordagens multimodais e modelos avançados como BERT e LLM. Também foram examinadas pesquisas sobre a análise de interações ilícitas em plataformas digitais e a aplicação de soluções forenses em ambientes criptografados, como o *WhatsApp*.

Por fim, a inclusão de estudos recentes em língua portuguesa reforçou a importância de corpora anotados, adaptação de domínio e avaliação criteriosa de modelos para contextos jurídicos e criminais. Em conjunto, esses achados demonstram a diversidade metodológica necessária para enfrentar a dinâmica do crime no ambiente digital.

Essa revisão reforçou a relevância do tema e a lacuna existente no contexto específico da perícia forense, justificando a pesquisa proposta. A partir deste contexto, o Capítulo 4 apresentará a Metodologia Proposta, detalhando o fluxo de trabalho para a coleta, engenharia, rotulação e treinamento dos modelos de classificação de sentenças.

4

METODOLOGIA PROPOSTA

Este capítulo detalha os métodos empregados na pesquisa, desde a extração de de dados, limpeza (*datas cleaning*), rotulação, balanceamento dos dados, *fine-tuning*, até a geração e seleção do modelo preditivo mais eficiente para o projeto proposto.

A proposta inicial fundamentava-se na utilização de uma base de dados extraída de um arquivo de imagem ("*.bin*"). Para esse fim, foram desenvolvidos scripts específicos, responsáveis por realizar a extração do conteúdo e sua organização para posterior utilização.

Todavia, quando do início dos experimentos, a base de dados foi trocada deste arquivo de imagem para diretórios contendo um arquivo de banco de dados (SQLight) gerado pela Polícia Científica do Estado do Paraná (PCP/PR) a partir de seu relatório de perícia (*.pdf*), além dos arquivos referentes à imagens e áudios trocados nas mensagens.

Para tal, optou-se pela utilização de modelos de classificação de sentenças, com o objetivo de verificar se uma determinada mensagem textual apresenta indícios ou conotação associada a práticas criminosas.

Na sequência, a Figura 4.1 apresenta o processo utilizado para desenvolvimento deste projeto.

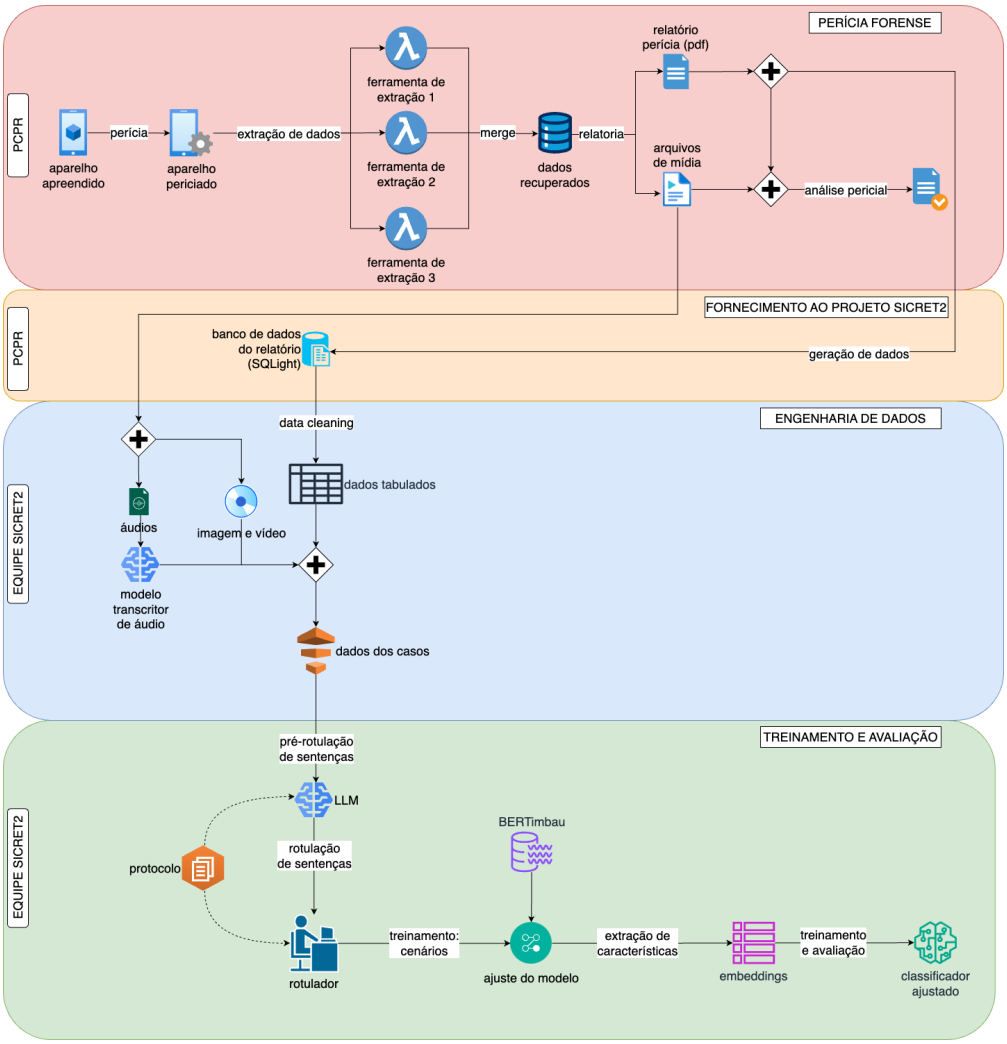


Figura 4.1: Fluxo do projeto SICRET II
Elaborado pelo autor

As primeiras etapas deste fluxograma, Figura 4.1, representam o fluxo de perícia forense, que, para este projeto, atua como fornecedores de dados. Estes dados foram selecionados e filtrados para a modelagem de um sistema classificador de sentenças que visa agilizar o processo de identificação de conversas e trechos com potencial viés criminoso.

Na sequência do método proposto, apresentam-se as demais etapas do método proposto, detalhadas ao longo deste capítulo, a saber: engenharia de dados, transcrição de áudios, organização das sentenças, rotulação — realizada tanto por meio de modelos LLM quanto manualmente pelo pesquisador — e, por fim,

o ajuste e treinamento dos modelos em diferentes cenários experimentais.

4.1 Extração de Dados

Para extração de dados dos celulares periciados a polícia acaba utiliza diferentes ferramentas de extração de dados, entre elas o Cellebrite (PARANÁ, 2023) e o Android Debug Bridge (ADB) (VIEIRA; CRUZ, 2016). Além disso, faz-se necessário o emprego de mais de um software, de modo a maximizar a recuperação de dados, suprindo eventuais limitações de cada sistema individualmente.

De posse dessas extrações, a PCPR utiliza o sistema Mobile Merge – sistema próprio, desenvolvido pela própria polícia – para concatenação desses dados e geração de um relatório único.

A partir do relatório pericial em formato ".pdf" — que reúne as informações textuais e as referências a áudios, imagens e vídeos extraídos do aparelho analisado — a PCPR, por meio de aplicação de desenvolvimento próprio, gera um banco de dados (SQLite) contendo os textos do relatório e seus respectivos números de página.

Além disso, todos os arquivos de mídia e ".pdf" referenciados ficam organizados em um diretório auxiliar a fim de possibilitar sua análise. Todos esses elementos compõem um "caso" qual se refere ao celular em questão que estaria sob análise pericial.

Cabe salientar que a presente pesquisa não possui acesso físicos aos aparelhos periciados, nem tampouco aos seus dados brutos, sendo disponibilizado apenas os arquivos relativos ao laudo pericial, aos resultantes de sua conversão em banco de dados e dos arquivos de mídia (áudio, imagem e vídeo) do referido caso.

Convém ressaltar que, para o treinamento dos modelos, além da execução em ambiente de processamento *on-premise*, foram utilizados exclusivamente os conteúdos das mensagens, sem qualquer necessidade de identificação dos indivíduos proprietários dos dispositivos.

4.2 Engenharia de Dados

Devido ao fato de o banco de dados utilizado neste projeto ser gerado a partir de um arquivo de texto – relatório em ".pdf" – houve a necessidade de organizar seu conteúdo, primeiramente identificando os dados do proprietário do celular,

bem como dos receptores das mensagens, quer individual, quer por grupos de troca de mensagens.

Também foram selecionadas apenas conversas trocadas no aplicativo *Whatsapp*[1] (foco desta pesquisa), dada a identificação de padronagem na organização dos textos, além de ser o aplicativo com maior utilização no Brasil na atualidade (BARCELOS; MARQUES; FILHO, 2019).

A partir dessas premissas, as mensagens foram extraídas dos texto com uso de técnicas de expressões regulares (*Regular Expressions* (REGEX)), que consistem na identificação de texto a partir de um padrão delimitante de caracteres.

A título elucidativo, abaixo demonstramos uma aplicação – com dados fictícios – de REGEX utilizada durante o processo de *data-cleaning*, no caso específico, aplicado a fim de identificar o nome do grupo ou do indivíduo; salientando que por se tratar de dados não estruturados (apesar de estarem armazenados em banco de dados, o conteúdo a ser analisado não textos corridos, sem qualquer estruturação) a organização dos dados teve de ser feita a partir de identificação de padrões estruturantes:

Texto do banco de dados:

```
identificador: 999999999999999999@g.us =
sicret2:simulação primeiro registro:
```

REGEX aplicado:

```
identificador: \d+@(?:g\.us|s\.whatsapp\.net) =
(.+?) primeiro registro:
```

- "**`identificador:`** ": Este trecho do regex busca literalmente a palavra "`identificador`" seguida de dois pontos e um espaço. Ele só corresponderá a strings que contenham exatamente essa sequência de caracteres.
- "**`\d+`**": Esta parte corresponde a uma ou mais ocorrências de dígitos (`0-9`). O sinal de "+" significa "um ou mais", e `\d` é uma classe de caractere que representa qualquer dígito numérico.
- "**`@`**": Corresponde literalmente ao caractere arroba (`@`).

[1]WHATSAPP. WhatsApp. Disponível em: <https://www.whatsapp.com/>. Acesso em: 19 ago. 2024.

- "`(?:g\.us|s\.whatsapp\.net)`": Esta é uma expressão não capturante (indicada pelo ponto de exclamação (?): no início), que significa que ela corresponde ao texto, mas não armazena o texto correspondido para uso posterior. A expressão pode corresponder a "`g.us`" ou "`s.whatsapp.net`". O ponto final (`.`) é um caractere especial em REGEX que corresponde a qualquer caractere único, exceto uma nova linha, então ele é escapado aqui (`\.`) para corresponder literalmente a um ponto.

- "`=`": Corresponde literalmente à sequência de caracteres " = " (espaço, igual, espaço). Utilizado para separar o identificador do valor que será capturado a seguir.

- "`(.+?)`": Esta é uma expressão de captura que corresponderá a uma ou mais ocorrências de qualquer caractere, exceto a nova linha, o mínimo de vezes possível. O ponto final (`.`) corresponde a qualquer caractere (exceto nova linha), o sinal de adição (+) indica "um ou mais", e o ponto de exclamação (?) torna a quantificação "preguiçosa" (ou seja, tenta corresponder o mínimo possível de caracteres que ainda permitem que o REGEX geral corresponda).

- "`primeiro registro:`": Corresponde literalmente a sequência de caracteres " `primeiro registro:`" incluindo o espaço inicial.

Após a identificação das mensagens efetivamente trocadas, os dados foram organizados de forma a conferir maior clareza e fluidez à leitura. Para tanto, as mensagens foram ordenadas segundo o registro temporal (data e hora) e agrupadas conforme o emissor e o receptor — indivíduo ou grupo —, não com o intuito de identificar a persona, mas de assegurar a correta sequência das interações.

Outros pontos de atenção dizem respeito às informações constantes no relatório pericial, tais como: se a mensagem foi excluída pelo usuário, se foi excluída e posteriormente recuperada na extração de dados, ou ainda se toda a conversa havia sido removida. Embora esses elementos não exerçam impacto direto sobre o presente projeto, optou-se por mantê-los em virtude de sua relevância para a atividade pericial.

A fim de incrementar e trazer maior robustez às análises dos dados, não apenas as mensagens textuais foram utilizadas como também as mensagens de voz enviadas pelos usuários. Para isso, foi necessária a transcrição desses

áudios para texto com utilização de modelos específicos para essa tarefa, como Vosk (MARTINS et al., 2021) e Whisper (RADFORD et al., 2023), sendo que o resultado dessas transcrições foram inseridos nos dados tabulados como sendo o texto da mensagem em questão. Vale destacar, que foi avaliada a eficácia e eficiência desses transcritores, a partir de modelos de diferentes tamanhos, a saber: Vosk (*small* e *large*) e Whisper (*base*, *tiny*, *small* e *medium*).

Ao final deste processo, os dados foram tabulados gerando um arquivo para cada caso (aparelho periciado), contendo os seguintes elementos:

Tabela 4.1: Detalhes da Mensagem (dados fictícios)

Campo	Descrição	Exemplo
date_time	data e hora	2021-05-18T22:33:28
sender_name	nome do remetente	fulano de tal
sender_phone	número do celular do remetente	554199999999
sender_type	tipo do remetente [proprietário do aparelho | participante]	participante
recipient	destinatário [indivíduo | grupo]	grupo de drogas
text	texto	vende um bagulho? qualquer droga aih
page_number	número da página do relatório	2
item_number	número do item do relatório	1
case	nome do caso	caso2
deleted_conversation	conversa deletada	nao
obs	observação	-
goldkeys_found	gold keys encontradas	[bagulho, droga]

Fonte: Elaborado pelo autor.

Observa-se que, no contexto desta pesquisa, a equipe pericial da PCPR forneceu um conjunto de palavras-chave de referência, previamente definido, denominado *gold keywords* (CAMPOS et al., 2020). Essas palavras, selecionadas por sua alta relevância na identificação de indícios de práticas criminosas — especialmente aqueles associados ao tráfico de drogas —, foram utilizadas como parâmetro inicial de comparação na avaliação dos modelos.

4.3 Transcrição dos dados

Como citado anteriormente, as mensagens de áudio foram transcritas e inseridas no respectivo contexto da conversa analisada, tendo sido testados os seguintes modelos transcritores:

- Vosk: *small* e *large*;
- Whisper: *base*, *tiny*, *small* e *medium*

Tendo em vista a inviabilidade da transcrição manual dos 25.768 áudios contidos na base de dados da pesquisa. Para avaliação da eficácia e eficiência dos modelos, foram utilizados áudios da obra Clube de Poesia (1995). No total, cinquenta áudios foram gerados pelo próprio pesquisador, utilizando sua voz e recursos próprios, esses originários de um projeto comunitário desenvolvido pela presente universidade.

Complementarmente, essa mesma avaliação foi feita a partir de dados reais, situação relevante devido as circunstâncias apresentadas por áudios gerados em ambientes não controlados, tendo sido selecionados, aleatoriamente, 5 áudios para cada um dos 9 casos analisados, totalizando 45 arquivos.

Os resultados foram avaliados considerando a eficácia, medida pelo *F1-Score* do teste *rouge*, conforme Equação 4.3, que compara palavra por palavra entre o texto original (referência) e o transcrito (gerado), e a eficiência, levando em conta o tempo de processamento (segundos).

$$\text{ROUGE-1}_{\text{recall}} = \frac{|\text{Unigrama}(C) \cap \text{Unigrama}(R)|}{|\text{Unigrama}(R)|} \tag{4.1}$$

$$\text{ROUGE-1}_{\text{precision}} = \frac{|\text{Unigrama}(C) \cap \text{Unigrama}(R)|}{|\text{Unigrama}(C)|} \tag{4.2}$$

$$\text{ROUGE-1}_{F1} = \frac{2 \times \text{ROUGE-1}_{\text{precision}} \times \text{ROUGE-1}_{\text{recall}}}{\text{ROUGE-1}_{\text{precision}} + \text{ROUGE-1}_{\text{recall}}} \tag{4.3}$$

- R = texto de referência (*reference text*)
- C = texto gerado (*candidate text*)
- $|\,\text{Unigrama(X)}\,|$ = número de unigramas no texto X

A seleção do modelo foi realizada considerando, simultaneamente, o maior valor de *F1-Score* e o menor tempo de processamento. Para essa comparação, aplicou-se o teste estatístico de Nemenyi sobre essas duas variáveis. A escolha final recaiu sobre o modelo com melhor desempenho global.

Quanto ao *F1-Score*, importa ressaltar que sua escolha se deu pelo fato de combinar os resultados da precisão (*precision*) e revocação (*recall*) – (KUBAT, 2017), conforme apresentado na Equação 4.3.

4.4 Rotulação de Dados

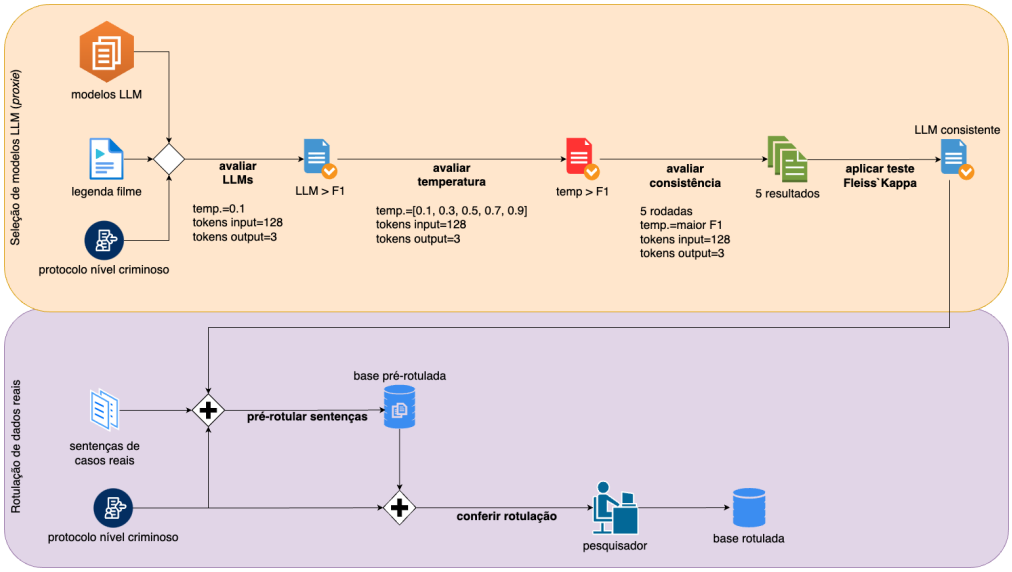
Durante o processo de planejamento do método de avaliação de modelos classificadores de sentenças, foi percebida a necessidade de aplicação de técnica de *fine-tuning* com intuito de otimizar a compreensão dos modelos especificamente para o contexto abordado nesta pesquisa.

Para isso, foi necessário gerar uma base rotulada capaz de servir de insumo para esse processo de ajuste fino.

Assim, foram selecionados pela perícia 9 casos reais de dispositivos apreendidos que possibilitaram a criação dessa base, consistindo num total 392.416 sentenças, 2.961.033 palavras.

O desenvolvimento desta etapa do projeto exigiu a execução de algumas atividades de seleção de modelo LLM, pré-rotulação de dados e conferência humana, como detalhado nesta seção e ilustrado na Figura 4.2.

Figura 4.2: Fluxo de rotulação de dados do projeto SICRET II



Fonte: Elaborado pelo autor.

A classificação dessas sentenças foi executada, quer pelos modelos de linguagem, que pela avaliação do pesquisador, respeitando os dois tipos de classificação, conforme protocolo abaixo:

Nível de envolvimento criminoso

- **Classe 0 (zero) - Sentença sem conteúdo criminoso**
  - Sentença: "Fala mano, blz?"
  - Sentença: "Pode pegar o Marquinhos em casa?"
  - Sentença: "kkkkkkkkkkkkkkkkkkk"
- **Classe 1 (um) - Sentença com indício de envolvimento criminoso**
  - Sentença: "Os cara estam com armas, dando tiro"
  - Sentença: "Ei, me vende um bagulho aih!"
  - Sentença: "Vc quer fumar uma maconha, vai um baseado aih?"

4.4.1 Seleção de modelos LLM para pré-rotulação de dados

Considerando o elevado volume de dados a serem analisados — 392.416 sentenças e um total de 2.961.033 palavras —, bem como as limitações de prazo e de recursos humanos disponíveis, esta pesquisa recorreu à utilização de modelos de LLM para auxiliar na fase de rotulação. Esses modelos foram empregados para realizar a pré-rotulação das sentenças, cuja consistência foi posteriormente avaliada e validada pelo pesquisador.

Com o objetivo de avaliar o potencial dos modelos de LLM na tarefa de rotulação de dados, foram selecionados aqueles que atendessem aos seguintes critérios:

- modelo aberto, passível de ser processado localmente;
- versões mais atuais de modelos disponibilizados;
- multilíngue ou voltado à língua portuguesa;
- tamanho do modelo, compatível com processamento em computador pessoal.

Assim, foram selecionados, inicialmente, os seguintes modelos:

- Meta LLAMA3 8B[2];

[2]meta-llama/Meta-Llama-3-8B

- LLAMA3 8B Dolphin Portuguese, v0.3[3];
- LLAMA3 8B Dolphin Portuguese, v0.4[4];
- LLAMA3 8B Instruct Portuguese, v0.3[5];
- LLAMA3 Portuguese Tom Cat 8B Instruct[6]

Embora versões mais recentes, em teoria, apresentariam melhorias em relação às anteriores, optou-se por incluir diferentes variantes e *releases* para fins de comparação empírica. Isso se deve ao fato de que tais modelos, apesar de compartilharem a mesma base (LLAMA3), apresentam diferenças em aspectos relevantes, tais como: corpus de adaptação, técnica de instrução (instruct vs. dolphin), ajustes de *fine-tuning* realizados por comunidades distintas. Dessa forma, a experimentação buscou verificar, na prática, se as versões mais recentes superavam de forma consistente as anteriores no contexto específico da rotulação forense em língua portuguesa.

A avaliação da capacidade de rotulação foi testada utilizando, como *proxy*, as legendas do filme "Cidade de Deus" (MEIRELLES, 2002), obra escolhida pelo fato de retratar cenas relacionadas a temas como envolvimento criminoso e drogas.

Para possibilitar uma maior contextualização, cada sentença a ser rotulada foi considerada a como a concatenação de até 128 *tokens*, sendo o contexto geral da sentença em análise composto pela sentença anterior e posterior, quando houver.

Nessa experimentação, foram utilizadas os seguintes parâmetros:

- temperaturas: [0.1, 0.3, 0.5, 0.7, 0.9];
- número de tokens gerados: 3

A verificação do desempenho desses modelos se deu pela comparação entre a classificação feita pelo LLM e a rotulação humana efetuada pelo pesquisador, considerando os valores do *F1-Score*.

[3] adalbertojunior/Llama-3-8B-Dolphin-Portuguese-v0.3
[4] adalbertojunior/Llama-3-8B-Dolphin-Portuguese-v0.4
[5] adalbertojunior/Llama-3-8B-Instruct-Portuguese-v0.3
[6] rhaymison/Llama-3-portuguese-Tom-cat-8b-instruct

Vale salientar que para a primeira comparação entre os modelos de linguagem, foi utilizada apenas a "*temperatura = 0.1*", sendo os demais valores utilizados para verificação de desempenho do LLM que obtivera maior *F1-Score*, no intuito de identificar qual parâmetro potencializaria a análise do modelo.

Como *prompt* de entrada para a rotulação do modelo de linguagem, foi considerado protocolo juntamente com o contexto geral supracitado.

Ademais, com o objetivo de verificar a regularidade do LLM com melhor desempenho, considerando sua característica estocástica e visando maior segurança na seleção do modelo, este foi submetido a 5 (cinco) rodadas de testes. Os testes seguiram o mesmo protocolo e os parâmetros previamente definidos, utilizando a mesma base de dados. Foram avaliadas a consistência dos resultados do *F1-Score* e a qualidade da rotulação, por meio do teste de concordância de Fleiss' Kappa (MOONS; VANDERVIEREN, 2023), conforme os níveis de concordância apresentados na Tabela 4.2.

Tabela 4.2: Fleiss'Kappa: parâmetros de interpretação de resultados

Estatística K	**Nível de Concordância**
<0.00	pobre
0.00-0.20	leve
0.21-0.40	justa
0.41-0.60	moderada
0.61-0.80	substancial
0.81-1.00	quase perfeita

Fonte: Elaborado pelo autor, adaptado de Landis & Koch (1977).

4.4.2 Pré-rotulação de dados reais com LLM

Após a etapa de comparação dos modelos LLM, cujos resultados são discutidos na Seção 5.4.1, o modelo selecionado foi usado para pré-rotulação da base de dados real da pesquisa.

Posteriormente, essa base foi cuidadosamente revisada pelo pesquisador, sentença a sentença, a fim de garantir a precisão das rotulações e a adequação ao objetivo do estudo.

Para garantir a integridade semântica e o contexto das mensagens, estas foram agrupadas com base nos integrantes de cada conversa, podendo ser indi-

viduais ou em grupos. Além disso, foram organizadas cronologicamente pela data de envio, preservando o sentido original das conversas. A estruturação também considerou a concatenação sequencial de até "*512 tokens*", respeitando o limite imposto pelo modelo, para maximizar a eficiência e a compatibilidade com o processo de análise.

Vale salientar que, para a presente avaliação, foi necessário aplicar a técnica de quantização em "*4 bits*", uma abordagem de compressão que reduz significativamente os requisitos computacionais do modelo, tornando possível sua execução em hardware com recursos limitados.

Conforme argumentado por Dettmers & Zettlemoyer (2023), essa técnica pode ocasionar certa perda de precisão durante a inferência, contudo, o processo de quantização foi necessário dadas as restrições impostas pela Graphics Processing Unit (GPU) fornecida pela PCPR para experimentação.

A rotulação realizada pelo modelo foi enriquecida por meio da contextualização das sentenças anteriores e posteriores, sempre que disponível.

Essa abordagem se justifica pela importância do contexto na interpretação de mensagens, especialmente em tarefas relacionadas à análise forense.

Contudo, para a primeira sentença de cada conversa, não havia contexto anterior disponível, assim como a última sentença não contava com um contexto subsequente.

Apesar dessas limitações, o processo buscou assegurar que as rotulações refletissem com precisão o significado original das mensagens no escopo do estudo.

Convém destacar que a avaliação da rotulação realizada pelo LLM foi conduzida por meio da métrica *F1-Score*, tomando-se como referência a média ponderada dos resultados obtidos pelos diferentes casos reais analisados, conforme Equação 4.4. Essa avaliação foi feita em comparação direta com a rotulação manual realizada pelo pesquisador, considerada neste estudo como padrão de referência (*baseline*).

$$\bar{F}_1^{\text{ponderada}} = \frac{\sum_{i=1}^{n} F_{1,i} \cdot w_i}{\sum_{i=1}^{n} w_i} \tag{4.4}$$

Em particular, as componentes da Equação 4.4 são:

- $F_{1,i}$: valor do *F1-Score* de cada caso;

- w_i: número de amostras em cada grupo (tamanho do caso);
- n: número total de casos.

Vale ressaltar essa opção metodológica, do uso da média ponderada pelo número de amostras ao invés da média simples, deve-se ao fato de que os casos avaliados apresentam tamanhos bastante distintos, tanto em número de sentenças quanto em relação ao desbalanceamento das amostras, tal como detalhado na Tabela 5.8.

Assim, a simples média aritmética poderia atribuir o mesmo peso a casos com características bastante distintas. Isso se tornaria especialmente problemático em cenários nos quais o modelo precisou classificar um volume maior de amostras, enfrentando, portanto, mais dados desconhecidos, ao mesmo tempo em que recebeu um menor *input* de treinamento, ou seja o modelo precisou ter um maior poder de generalização.

Essa discrepância poderia levar a uma visão distorcida do desempenho. Ao utilizar a ponderação pelo número de sentenças de cada caso, buscou-se assegurar que a métrica global refletisse de forma mais fiel a performance do modelo em relação ao conjunto de dados como um todo.

4.5 Balanceamento dos Dados

Caso seja identificado um desbalanceamento no conjunto de dados, ou seja, quando há um número significativamente maior de amostras em uma classe em relação a outra(s), é fundamental aplicar técnicas de balanceamento. Isso evita que o modelo aprenda de forma desproporcional, favorecendo a classe majoritária e, possivelmente, negligenciando a classe minoritária, o que poderia introduzir viés no aprendizado do modelo (WONGVORACHAN; HE; BULUT, 2023).

Nesta pesquisa foi adotada a abordagem baseada em dados, especificamente o *oversampling* e o *undersampling*. Optando-se pela utilização das técnicas *Random Oversampling* (ROS) e *Random Undersampling* (RUS) como estratégias de balanceamento de dados.

Embora existam métodos mais sofisticados, a escolha recaiu sobre as abordagens aleatórias por três motivos principais: (i) ampla utilização em estudos de referência, o que garante comparabilidade dos resultados; (ii) menor custo

computacional, aspecto relevante diante do grande volume de sentenças analisadas; e (iii) alinhamento ao objetivo central desta etapa, que não foi o de explorar exaustivamente todas as técnicas de balanceamento, mas sim verificar de forma controlada o impacto do uso de abordagens clássicas de *oversampling* e *undersampling* sobre o desempenho dos modelos testados.

Assim, buscou-se privilegiar a clareza metodológica e a reprodutibilidade, deixando a investigação de técnicas mais complexas como perspectiva para trabalhos futuros.

Na técnica do *oversampling*, são criadas amostras da casse minoritária até que se atinja um equilíbrio com a classe majoritária. Por outro lado, a técnica *RUS* atua reduzindo o número de amostras da classe majoritária até equipará-la à minoritária.

Essas duas técnicas são aplicadas a fim de verificar se a ampliação ou a redução amostral pode gerar algum efeito significativo na eficácia dos modelos.

4.6 Representação

Nesta seção, retomamos o conceito de *embeddings*, abordado na Subseção 2.5.4, onde apresentamos os *Transformers* como modelos que geram representações vetoriais de palavras. Os *embeddings* são obtidos a partir de um treinamento prévio em grandes conjuntos de dados textuais *corpora* com o objetivo de capturar características sintáticas e semânticas das palavras em diferentes contextos (SOUZA, 2020).

No contexto desta pesquisa, fora utilizado um modelo pré-treinado voltado para a língua portuguesa – seus respectivos *embeddings* – aplicando-o na classificação de sentenças relacionadas a atividades criminosas, tendo sido utilizado o modelo:

- **BERTimbau** [7];
  - *base*: 110 milhões de parâmetros;

O fato desse modelo ser dedicado à língua portuguesa, o torna particularmente relevante para esta pesquisa, já que o objeto de estudo está intrinsecamente relacionado a dados textuais nesse idioma. Além disso, por ser

[7] Souza, Nogueira & Lotufo (2020)

um modelo moderno e disponibilizado recentemente, ele incorpora avanços arquiteturais que possibilitam maior precisão em tarefas de processamento de linguagem natural.

A avaliação proposta busca verificar a capacidade desses *embeddings* de capturar as nuances sintáticas e semânticas do domínio forense, considerando tanto o pré-treinamento geral quanto o impacto de ajustes específicos, como o *fine-tuning*, para a tarefa de classificação de sentenças.

Essa técnica pode ser conduzida em dois grandes enfoques: supervisionado, quando se utiliza um conjunto de dados rotulado, e não supervisionado, quando o processo se baseia em dados não rotulados

Especificamente no caso do ajuste supervisionado (*fine-tuning*), Bilal & Almazroi (2023) apontam que a aplicação desta técnica permitiu melhoras no desempenho do modelo avaliado. Vale ressaltar que, nesse mesmo estudo, foram avaliados ajustes para seis comprimentos de sequências.

Para a presente pesquisa foi padronizada a utilização da quantidade máxima de *tokens* permitido pelo modelo – 512, no caso – considerando a necessidade de maior contexto possível para que fosse possível que o modelo compreendesse qualquer início de envolvimento criminoso nas mensagens analisadas. Essa escolha se justifica pelo fato de os textos não seguirem uma estrutura gramatical formal, apresentando frases truncadas, múltiplos temas simultâneos e, muitas vezes, sequências desconexas, além do uso de neologismos.

Com relação à técnica de ajuste não supervisionado (sem a necessidade de dados rotulados) – *Masked Language Model* (MLM) – de modelos de pré-treinados, como o caso do BERTimbau, tem o objetivo de adaptar o modelo à tarefas específicas (LIANG; LIANG, 2024), ajustando-os para que sejam capaz de prever os *tokens* mascarados, a partir do contexto de *tokens* não mascarados, permitindo que o modelo compreenda o contexto bidirecional que lhe foi passado.

Também, deve-se considerar que, embora haja o modelo BERTimbau *large* (com 330 milhões de parâmentros), não foi possível sua utilização devido as limitações computacionais vivenciadas durante a pesquisa, uma vez que todo processamento teve de ser realizado com equipamento fornecido e dentro das instalações da Polícia Civil do Estado do Paraná (PCPR).

4.7 Variáveis de Treinamento

Nesta etapa, definiu-se a forma de representação textual utilizada como entrada para os modelos, com o objetivo de explorar diferentes estratégias de extração de informação a partir do conteúdo das mensagens. Para isso, foram construídas três variáveis de treinamento: *sentence*, *keyphrase* e *kp + sent*. Essas variáveis foram geradas a partir das conversas extraídas dos dispositivos periciados, respeitando as restrições de tamanho impostas pelo processamento dos modelos baseados em *Transformers*.

A variável "*sentence*" corresponde ao texto-base utilizado na modelagem, obtido pela concatenação das mensagens trocadas em uma mesma conversa, preservando sua ordenação temporal e o limite máximo de 512 *tokens*.

Já a "*keyphrase*" representa uma síntese do conteúdo da *sentence*, construída por meio da extração de palavras-chave (ou frases-chave). O método de extração foi definido de forma condicional ao rótulo atribuído à instância, visando refletir a disponibilidade de conhecimento humano nos exemplos positivos e, ao mesmo tempo, automatizar o processo para os exemplos negativos.

Por fim, a variável "*kp + sent*" foi definida como a concatenação das variáveis *keyphrase* e *sentence*, formando uma representação híbrida que combina uma síntese semântica (palavras-chave) com o conteúdo integral da conversa.

Essa concatenação foi realizada com a *keyphrase* posicionada antes da *sentence*. Essa decisão metodológica foi necessária porque a *sentence* pode atingir o limite de 512 *tokens*; assim, caso a *keyphrase* fosse adicionada ao final, haveria risco de truncamento e consequente descarte das palavras-chave durante o processamento, reduzindo sua contribuição para o modelo.

4.8 Fine-tuning

Tendo em vista que o modelo de linguagem citado na seção anterior possui a característica de ser generalista, torna-se necessário adaptá-lo para tarefas específicas, como a classificação de sentenças relacionadas a atividades criminosas. Essa adaptação é relevante para que o modelo consiga captar com precisão os padrões linguísticos específicos de contextos forenses.

Nesse sentido, este trabalho propõe o emprego da técnica de ajuste fino (*fine-tuning*), que visa permitir que os modelos aprendam características específicas

da linguagem associada a práticas criminosas. O ajuste fino consiste em expor os modelos a um conjunto de dados, anotados ou não, a depender da técnica utilizada, possibilitando que eles ajustem seus pesos internos para capturar nuances e padrões relevantes para a tarefa.

Convém ressaltar que este projeto contempla a aplicação e a avaliação de desempenho de duas abordagens de ajuste de modelos pré-treinados: o *Fine-Tuning*, de natureza supervisionada, realizado com dados rotulados pelo pesquisador; e o *Masked Language Model* (MLM), de natureza não supervisionada, que dispensa a utilização de dados rotulados.

Dessa forma, esta pesquisa se propõe a aplicar o ajuste fino ao modelo mencionado anteriormente. O objetivo é aprimorar sua capacidade de classificação por meio da atualização de seus pesos e, consequentemente, gerar novos *embeddings* mais representativos e específicos para o contexto forense. Esses *embeddings* poderão, então, ser utilizados para melhorar a precisão e a robustez dos sistemas de análise de linguagem em aplicações práticas.

4.9 *Hyperparameter Optimization* (HPO)

Com vistas a robustecer o desempenho dos modelos durante a etapa de *fine-tuning*, esta pesquisa aplicou a técnica de *Hyperparameter Optimization* (HPO) utilizando a biblioteca Optuna (AKIBA et al., 2019). O objetivo central foi automatizar a busca pelas configurações que melhor se adaptassem às especificidades dos dados forenses, superando as limitações e a ineficiência do ajuste manual.

A metodologia de HPO foi aplicada de forma exaustiva para cada combinação de tipo de balanceamento (RUS e ROS) e para os diferentes cenários experimentais propostos que faziam uso do modelo pré-treinado BERTimbau, conforme disposição na Tabela 4.3. Essa abordagem garantiu que cada subconjunto de dados e cada arquitetura de treinamento recebessem um ajuste refinado e personalizado.

Tabela 4.3: Cenários com aplicação do HPO

		Variáveis de treino		
Balanceamento	**Modelo / Extrator**	**sentence**	**keyphrase**	**kp + sent**
ROS	FineTune BERT	✓	✓	✓
	MLM	✓	✓	✓
RUS	FineTune BERT	✓	✓	✓
	MLM	✓	✓	✓

Fonte: Elaborado pelo autor

Os hiperparâmetros submetidos ao processo de otimização e seus respectivos intervalos de busca foram:

- Taxa de aprendizado (*learning rate*): intervalo contínuo de 1e-5 a 5e-5;
- Épocas (*epoch*): intervalo discreto de 2 a 5;
- Tamanho do lote (*batch size*): 8 ou 16 amostras por lote.

4.10 Classificação

A classificação de sentenças, objetivo final desta pesquisa, se deu com o emprego dos algoritmos descritos no Capítulo 2, a saber: SVM, Naive-Bayes, MLP, Random-Forest, XGBoost e o SentenceBERT.

Esses algoritmos foram selecionados por representarem diferentes abordagens de aprendizado supervisionado: modelos lineares e probabilísticos (SVM e Naive-Bayes), redes neurais (MLP), métodos baseados em conjuntos (Random-Forest e XGBoost) e modelos de representação semântica baseados em *transformers* (SentenceBERT). Essa diversidade metodológica permitiu avaliar o desempenho da classificação sob múltiplas perspectivas e graus de complexidade.

Esse processo foi efetuado com aplicação de uma fase de treinamento dos modelos classificadores em combinação com os modelos de linguagem ajustados (*embeddings*), aplicando uma técnica de *cross-validation* por grupos – *Leave One Group Out* (LOGO) – apresentada na Seção 4.11.

O processo de classificação envolveu uma fase de treinamento em que os modelos foram alimentados com *embeddings* textuais gerados a partir dos modelos de linguagem. Para assegurar maior robustez estatística, aplicou-se a técnica de *cross-validation* por grupos (*Leave One Group Out* (LOGO)), conforme detalhado na Seção 4.11. Essa estratégia possibilitou avaliar a capacidade de generalização dos classificadores ao submetê-los a cenários em que cada grupo de dados é tratado como inédito em relação ao treinamento.

Uma vez obtido o modelo textual especialista, ajustado por meio do *fine-tuning*, procedeu-se à classificação das sentenças oriundas dos casos periciais. Nesse estágio, os classificadores foram responsáveis por atribuir rótulos às sentenças, distinguindo aquelas que apresentavam indícios de conotação criminosa daquelas que não apresentavam.

4.11 Validação Cruzada por Grupo

Com o objetivo de avaliar o desempenho dos classificadores frente a diferentes técnicas de processamento de linguagem, foram definidos 30 cenários experimentais. Cada cenário combina quatro elementos: o extrator de características empregado, a técnica de NLP aplicada, a variável de treinamento considerada e a estratégia de balanceamento de dados. Todos os cenários listados na Tabela 4.4 serão testados com os classificadores descritos na Seção 4.10.

Para a validação, adotou-se a técnica *Leave One Group Out* (LOGO), que busca reproduzir uma condição mais próxima da realidade. Nela, o modelo treinado deve realizar previsões sobre um conjunto de dados totalmente novo e desconhecido, simulando o contexto prático de aplicação.

Tabela 4.4: Cenários aplicados

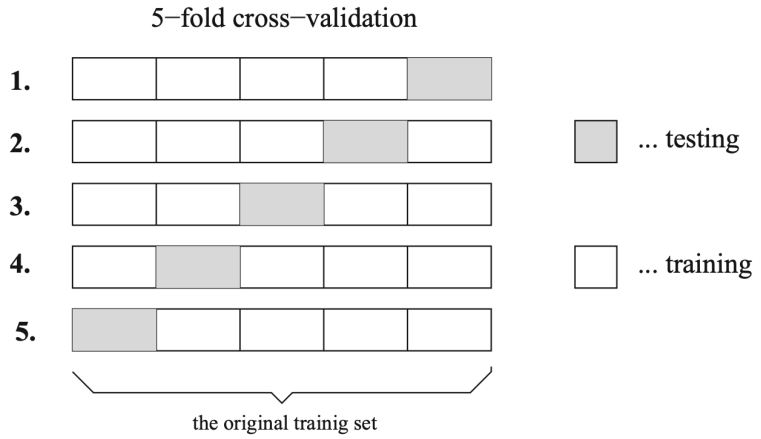
		Variáveis de treino		
Balanceamento	**Modelo / Extrator**	**sentence**	**keyphrase**	**kp + sent**
ROS	BERT Original	✓	✓	✓
	FineTune BERT	✓	✓	✓
	MLM	✓	✓	✓
	SBERT	✓	✓	✓
	TF-IDF	✓	✓	✓
RUS	BERT Original	✓	✓	✓
	FineTune BERT	✓	✓	✓
	MLM	✓	✓	✓
	SBERT	✓	✓	✓
	TF-IDF	✓	✓	✓

Fonte: Elaborado pelo autor

Essa validação por grupos consiste numa ideia semelhante ao *cross-validation* tradicional, porém ao invés de a troca ser feita a partir de um *range* – que por exemplo, pode ser de 10% – e sim por grupo, ou caso (aparelho de celular) no presente caso.

A estrutura demonstrada na Figura 4.3 elucida esse processo de validação, a amostra é processada "**N**" vezes, de maneira que, progressivamente, haja a troca na parte da base de dados destinada à validação, para que ao final o modelo possa ter aprendido as características e efetuado a validação de todas as amostras.

Figura 4.3: Exemplo de aplicação da técnica de *cross-validation*



Fonte: Kubat (2017).

Desta maneira, dos 9 casos reais presentes na amostra, cada iteração do experimento consiste em utilizar 8 casos para treinamento e 1 caso isolado para teste, obedecendo à lógica do método *Leave One Group Out* (LOGO). Esse processo é repetido de forma iterativa, garantindo que cada caso seja utilizado como teste uma única vez, enquanto os demais compõem o conjunto de treinamento.

A fim de identificar a combinação – modelo classificador e de linguagem – que obtivera o melhor resultado, foi realizada a avaliação com emprego da métrica *F1-Score*, a qual consiste levar em consideração os valores de sua precisão e revocação.

A Figura 4.4 ilustra, de forma esquemática, o fluxo de processamento adotado. Inicialmente, os dados completos são divididos conforme o critério de casos (grupos), sendo selecionado um único caso para teste e os demais para treino. O conjunto de treino é então submetido a técnicas de balanceamento (*undersampling* e *oversampling*), preservando a distribuição original do grupo de teste.

Na etapa de definição da **representação**, verifica-se se será utilizado o modelo BERTimbau pré-treinado e, em caso afirmativo, se este será ajustado por meio de *fine-tuning* supervisionado ou *Masked Language Model* (MLM). Isso resulta na geração de diferentes variações do modelo, que servirão como base para extração dos *embeddings*.

Na sequência, realiza-se a extração de características (*features*) tanto para os

dados de treino quanto para o grupo de teste. Essa fase obedece a mesma lógica apresentada na seção anterior, a possibilidade do uso de diferentes variáveis no treinamento, enquanto que para o teste restringe-se o emprego da variável *sentence*.

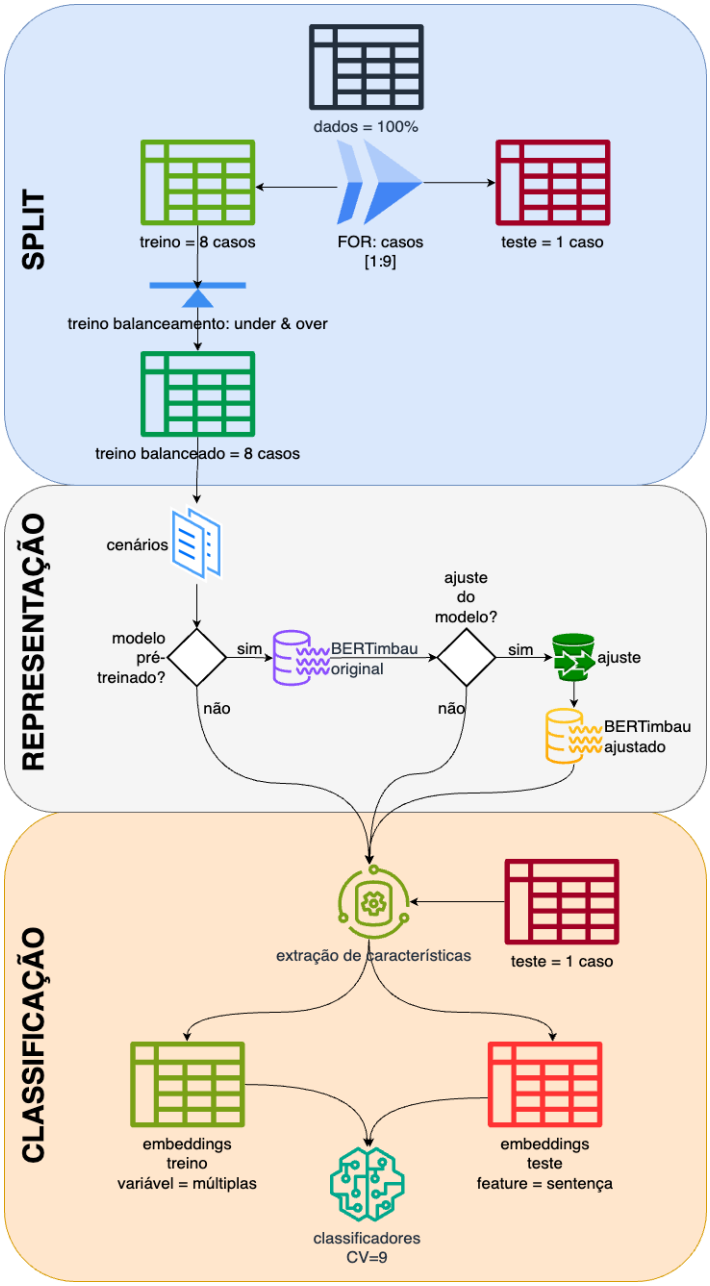
Além disso, os *embeddings* extraídos a partir do conjunto de treino são empregados no treinamento dos classificadores, assegurando que, a cada iteração do LOGO, seja realizada uma avaliação robusta e consistente do desempenho dos modelos.

Por fim, destaca-se que a avaliação das diferentes combinações de " **balanceamento | representação | classificação**" seguirá o mesmo critério adotado para o modelo baseado em LLM, conforme a Equação 4.4.

Assim, considerando a aderência do método *Leave One Group Out* (LOGO) à realidade da prática pericial, especialmente pela variação entre os casos analisados, estabeleceu-se o desempenho obtido por essa estratégia de validação, via média ponderada do *F1-Score* obtido nos casos pelo número de amostra do respectivo caso (Equação 4.4), como critério único para a escolha do modelo final.

Essa definição visa garantir que o modelo selecionado seja não apenas estatisticamente eficaz, mas também aplicável de forma robusta e realista no contexto forense, onde a diversidade dos casos impõe desafios que exigem capacidade de generalização.

Figura 4.4: Fluxo de processamento dos cenários de treinamento: *Leave One Group Out* (LOGO)



Fonte: Elaborado pelo autor.

4.12 Considerações Finais

Neste capítulo, foi detalhado o método de pesquisa, que visa gerar um modelo classificador de sentenças para auxiliar a perícia forense da Polícia Civil do Estado do Paraná (PCPR).

O processo se inicia com a extração e engenharia de dados (conversão de relatórios periciais para bases estruturadas), seguindo para a etapa de transcrição de áudios (avaliando modelos como Vosk e Whisper) e a crucial rotulação de dados reais, que incluiu a pré-rotulação com Grandes Modelos de Linguagem (LLM) e a validação manual.

Também foram definidas as técnicas de balanceamento de dados (ROS e RUS) e de representação textual. Por fim, foram estabelecidos 30 cenários experimentais para a classificação de sentenças, avaliados por meio da validação *Leave One Group Out* (LOGO), bem como para a aplicação da técnica de *Hyperparameter Optimization* (HPO).

A próxima etapa, detalhada no Protocolo Experimental, apresentará a infraestrutura utilizada, as características da base de dados forense, a seleção dos modelos transcritores e do LLM para rotulação, bem como a parametrização utilizada.

5

PROTOCOLO EXPERIMENTAL

Com a aplicação do método proposto, este capítulo apresenta uma visão detalhada dos insumos e procedimentos empregados na execução dos experimentos, bem como dos resultados obtidos. O objetivo é proporcionar uma compreensão clara e abrangente das etapas realizadas e dos dados utilizados ao longo do processo.

Tal qual abordado no capítulo anterior, o objetivo do método proposto foi a geração de um modelo de Processamento de Linguagem Natural (NLP) especialista na classificação de sentenças relacionadas a atividades criminosas.

Para isso, foram utilizadas como base de dados composta por mensagens de texto e áudio obtidas de celulares periciados pela PCPR, sem a utilização da identificação dos proprietários, de maneira a viabilizar o ajuste fino (*fine-tuning*) de um modelo base de processamento textual, adaptando-o às especificidades do domínio forense.

Vale salientar que, devido algumas limitações do equipamento utilizado para o processamento da classificação de sentenças utilizando o modelo LLM, confome Subseção 5.1.1, foi necessária a utilização da técnica de quantização, no caso para 4 *bits*. Conforme apontado por Huang et al. (2024), tal processo de quantização pode apresentar pequena queda na precisão do modelo – no caso analisado no estudo, o LLaMA3, do qual se deriva o modelo utilizado nesta experimentação – bem como um pequeno aumento de sua perplexidade, que segundo Tej (2024) é a medida de quão bem um modelo de linguagem prevê um trecho de texto.

Dessa forma, é plausível considerar que a utilização da quantização em 4 *bits*,

embora necessária para viabilizar os experimentos com os recursos disponíveis, possa ter impactado nos resultados obtidos.

Este capítulo apresenta, em primeiro lugar, as principais características do conjunto de dados utilizado. Em seguida, discute-se o desempenho dos modelos transcritores empregados. Na sequência, descreve-se a modelagem da pré-rotulação de dados com o uso de LLM, tanto em dados simulados quanto em dados reais. Posteriormente, é detalhado o protocolo experimental, incluindo a rotulação humana do conjunto de dados.

5.1 Infraestrutura de Execução dos Experimentos

Nesta seção, apresentam-se os detalhes do ambiente computacional empregado, incluindo a configuração do hardware e os recursos disponibilizados durante a execução dos experimentos.

5.1.1 Processamento com dados reais

No caso do processamento realizado com dados reais, adotou-se um ambiente computacional local, cedido pela Polícia Civil do Estado do Paraná (PCPR), de modo a garantir maior controle sobre os experimentos e reforçar os aspectos de segurança e sigilo das informações analisadas.

O experimento foi conduzido em um sistema com as seguintes especificações:

- **Computador**: Lenovo 30DJSBE200;
- **Processador**: Intel Core i7-13700K (Intel64 Family 6 Model 165 Stepping 3), operando a 3.01 GHz;
- **Memória RAM**: 32 GB;
- **Sistema Operacional**: Windows 11 Pro for Workstations (versão 10.0.22631);
- **Placa de Vídeo**: NVIDIA Quadro RTX 4000, equipada com 8 GB de memória GDDR6, 2304 núcleos CUDA, clock de 1545 MHz, largura de banda de 416.06 GB/s e suporte a DirectX 12.

Essa configuração foi responsável pela execução dos experimentos envolvendo dados reais, possibilitando tanto seu processamento quanto a confidencialidade necessária ao tratamento das informações.

5.1.2 Processamento com dados simulados

Os experimentos com dados simulador foram realizados no ambiente do *Google Colab*, com a seguinte configuração:

- **GPU**: **NVIDIA Tesla T4**, equipada com 15.36 GiB de memória total, driver 535.104.05 e versão 12.2 do CUDA;
- **CPU**: **Intel(R) Xeon(R) CPU @ 2.00GHz**, arquitetura x86_64, 2 núcleos, com caches L2 de 1 MiB e L3 de 38.5 MiB;
- **Virtualização**: ambiente executado por meio do hipervisor KVM.

Ressalta-se que a escolha desse ambiente para execução dos experimentos decorre do fato de, neste caso específico, serem utilizados apenas dados simulados, não havendo, portanto, qualquer restrição relacionada à sua sensibilidade.

5.2 Conjunto de dados

Para a execução deste projeto, a perícia selecionou 9 (nove) casos, cada um correspondendo a um aparelho celular submetido à análise pericial. Esses dispositivos continham dados relevantes para a investigação, incluindo mensagens de texto e arquivos de áudio, que serviram como insumos para os experimentos realizados.

Com o objetivo de fornecer uma visão detalhada sobre os dados utilizados, as principais características de cada caso estão apresentadas na Tabela 5.1. Importa destacar que, para o cálculo do número de palavras, foram consideradas as transcrições realizadas com o modelo Whisper, na versão *small* e que o número de sentenças retrata a estrutura original das mensagens, sem qualquer concatenação ou divisão.

Tabela 5.1: Dados dos casos analisados

Caso	Nº sentenças	Nº palavras	Nº áudios
1	11.689	118.698	1.749
2	1.124	23.141	562
3	131	1.196	69
4	151.178	1.301.023	3.454
5	38.254	236.030	2.561
6	13.123	133.209	3.360
7	4.900	29.727	269
8	103.673	645.010	11.200
9	68.344	472.999	2.544
Total	392.416	2.961.033	25.768
Média	43.602	348.845	2.863

Fonte: Elaborado pelo autor.

A Tabela 5.1 evidencia a heterogeneidade dos casos analisados, aspecto relevante para a avaliação do desempenho dos modelos. Observa-se, por exemplo, que o Caso 4 apresenta o maior volume de dados, com mais de 151 mil sentenças e 1,3 milhão de palavras, além de 3.454 áudios, configurando um cenário de alta densidade informacional.

Em contraste, o Caso 3 representa a menor amostra, com apenas 131 sentenças e 69 áudios, caracterizando um contexto bastante restrito. Há também casos intermediários, como o Caso 5 e o Caso 9, que apresentam tanto número expressivo de sentenças quanto quantidade relevante de áudios.

Outro destaque é o Caso 8, que, embora não seja o maior em número de sentenças, concentra a maior quantidade de áudios (11.200), o que reforça a importância da dimensão multimodal na análise.

Essa variabilidade — que vai desde amostras extremamente reduzidas até conjuntos massivos de dados — assegura que os experimentos realizados considerem diferentes cenários de complexidade.

Tal diversidade é importante pois simula os diferentes cenários a que são expostas as análises periciais. Logo, algo essencial para verificar a robustez dos modelos propostos diante de situações em que o volume e a qualidade do material periciado podem variar substancialmente.

5.3 Seleção do Modelo Transcritor

Nesta seção apresentamos os procedimentos utilizados na seleção do modelo transcritor a ser utilizado durante o projeto considerando seu desempenho a partir da assertividade e do tempo de processamento, utilizando os equipamentos descritos na Subseção 5.1.2, dos modelos transcritores com relação aos áudios utilizados durante a fase de teste.

Como abordado no Capítulo 2, esses transcritores – Whisper e Vosk – possuem diversos tamanhos; sendo que para este projeto foram selecionados:

- Whisper: *tiny*, *base*, *small*, *medium*;
- Vosk: *small*, *large*;

Dessa maneira, esses modelos foram avaliados conforme descrito no Capítulo 4, utilizando áudios da obra Clube de Poesia (1995) como *proxy*. Essa abordagem permitiu a criação de um cenário controlado para validar o desempenho dos modelos em termos de qualidade de transcrição e adequação ao domínio de aplicação.

Os resultados obtidos a partir dessa avaliação estão apresentados na Tabela 5.2, destacando métricas relevantes, como precisão, *F1-Score* e tempo de processamento, que fornecem uma visão abrangente do desempenho comparativo dos modelos.

Tabela 5.2: Avaliação de desempenho dos modelos transcritores

		Médias	
Modelo	**Tamanho**	***F1-Score***	**Tempo processamento (s)**
vosk	small	0,09	24,80
vosk	large	0,13	38,63
whisper	tiny	0,72	2,44
whisper	base	0,81	3,16
whisper	small	0,88	5,61
whisper	medium	0,86	10,01

Fonte: Elaborado pelo autor.

A Tabela 5.2 evidencia diferenças expressivas entre os modelos transcritores avaliados. Observa-se que os modelos da família Whisper superaram ampla-

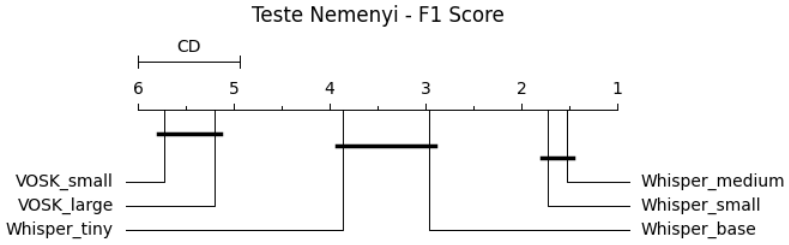
mente os modelos Vosk, tanto em termos de *F1-Score* quanto em tempo de processamento. Enquanto o Vosk apresentou desempenho muito baixo (*F1-Score* de 0,09 no modelo small e 0,13 no large), os modelos Whisper alcançaram valores significativamente superiores, variando de 0,72 (versão tiny) até 0,88 (versões small e medium).

Além disso, nota-se uma relação direta entre o tamanho do modelo Whisper e o tempo de processamento: quanto maior o modelo, maior o custo computacional, ainda que o ganho de desempenho se estabilize a partir das versões small e medium. Esse panorama inicial sugere a superioridade do Whisper em termos de qualidade de transcrição, ao mesmo tempo em que aponta para a necessidade de avaliar o equilíbrio entre acurácia e tempo de processamento na escolha do modelo mais adequado.

Assim, a fim de verificar se haviam diferenças estatisticamente significativas entre os resultados obtidos, foi aplicado o teste de Nemenyi, considerando duas métricas principais: o *F1-Score* e o tempo de processamento (em segundos). Este teste foi escolhido por sua adequação na comparação de múltiplos modelos, identificando quais apresentam diferenças relevantes de desempenho.

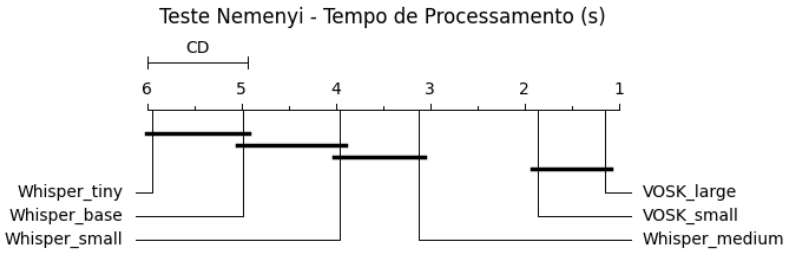
Os resultados da análise estão representados nas Figuras 5.1 e 5.2, que ilustram as comparações entre os modelos avaliados para cada métrica, destacando as possíveis diferenças estatísticas de forma visual e interpretativa.

Figura 5.1: Teste Nemenyi *F1-Score* - dados simulados



Fonte: Elaborado pelo autor.

Figura 5.2: Teste Nemenyi tempo de processamento em segundos - dados simulados



Fonte: Elaborado pelo autor.

A partir dos resultados obtidos, pode-se observar que o modelo Whisper Small apresentou o maior *F1-Score*, com um valor de 0,88, seguido de perto pelo Whisper Medium, com 0,86. No entanto, o tempo de processamento do Whisper Medium foi superior aos modelos Whisper Tiny e Base, demonstrando um desempenho menos eficiente nesse aspecto.

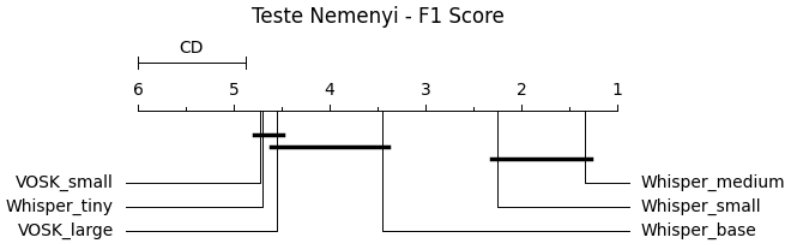
Os testes de Nemenyi revelaram que não há diferença estatisticamente significativa no *F1-Score* entre os modelos Whisper Small e Medium. Da mesma forma, não foram identificadas diferenças estatísticas no tempo de processamento entre os modelos Whisper Tiny, Base e Small.

A fim de avaliar a eficácia e a eficiência dos modelos em cenários reais - considerando amostras de dados utilizadas nesta pesquisa e as características específicas de áudios gravados em ambientes não controlados, como a presença de ruídos, neologismos e baixa qualidade de gravação - foram selecionados aleatoriamente 5 áudios para cada um dos 9 casos analisados, totalizando 45 arquivos.

Para a comparação da fidedignidade das transcrições, adotou-se como referência a transcrição manual realizada pelo pesquisador, seguindo a mesma metodologia aplicada anteriormente. Sendo assim, foram avaliados os resultados obtidos de *F1-Score* e tempo de processamento em segundo, utilizando os equipamentos apresentados na Subseção 5.1.1.

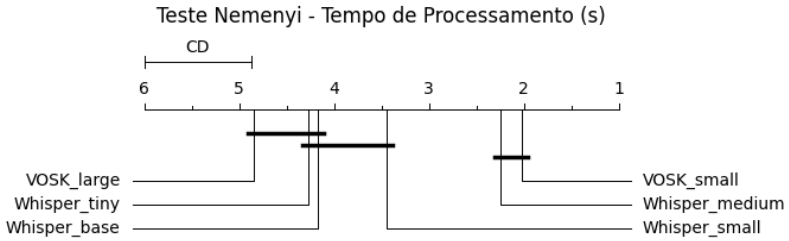
Os resultados dessa análise, apresentados nas Figuras 5.3 e 5.4, indicam que, em termos de eficácia, medida pelo índice F_1, não há diferença estatisticamente significativa entre os modelos Whisper Medium e Whisper Small. No entanto, em relação ao tempo de processamento, observa-se uma diferença entre esses dois modelos.

Figura 5.3: Teste Nemenyi *F1-Score* - dados reais



Fonte: Elaborado pelo autor.

Figura 5.4: Teste Nemenyi tempo de processamento em segundos - dados reais



Fonte: Elaborado pelo autor.

Especificamente, o modelo Whisper Medium apresentou o maior valor médio de F_1, com 0.80, seguido pelo Whisper Small, que obteve um valor médio de 0.69. No que se refere ao tempo de processamento, o Whisper Medium apresentou tempos mais elevados em comparação ao Whisper Small, sugerindo uma possível relação entre a maior precisão na transcrição e o custo computacional do modelo.

Com base nesses resultados, o modelo Whisper Small foi selecionado para as transcrições de todos os áudios dos casos reais do projeto.

Essa escolha se deve ao fato de o modelo apresentar um tempo de processamento reduzido, aliado a diferenças estatisticamente insignificantes em relação aos maiores valores de *F1-Score* observados, o que o torna mais adequado às necessidades do estudo.

5.4 Rotulação de dados

Esta seção apresenta o processo de rotulação dos dados que fundamenta a construção da base supervisionada utilizada nos experimentos. Considerando a natureza informal e heterogênea das mensagens, bem como o volume expressivo de sentenças, adotou-se um fluxo em camadas, combinando apoio automatizado e verificação humana.

Primeiramente, apresenta-se o critério de seleção do modelo LLM empregado como apoio inicial; em seguida, detalha-se a rotulação dos dados reais com LLM, utilizada como etapa preliminar de triagem e sugestão de rótulos; por fim, descreve-se a rotulação humana dos dados reais, responsável pela validação integral, correção de inconsistências e consolidação do rótulo final.

5.4.1 Seleção do Modelo LLM

Com o foco identificar o modelo LLM com maior viabilidade e desempenho na tarefa de classificação de sentenças como forma de auxiliar a rotulação de dados, foi realizada, inicialmente, uma simulação utilizando as legendas do filme Cidade de Deus (MEIRELLES, 2002).

Essa escolha desta base de treinamento se deve à riqueza linguística e à presença de construções textuais diversas, características que tornam o material adequado para testar a capacidade dos modelos em lidar com variações contextuais e semânticas.

Para favorecer a compreensão, pelo modelo de linguagem, do contexto em que a sentença está inserida, cada sentença (limitada a uma sequência de até 128 *tokens*) foi analisada pelo modelo considerando o contexto das sentenças anterior e posterior, quando disponíveis.

Essa abordagem busca assegurar que o modelo capture nuances importantes do significado, mesmo em situações onde a interpretação isolada de uma sentença possa ser ambígua ou incompleta.

Para essa primeira fase de experimentação do modelos selecionados, foram utilizadas os seguintes parâmetros:

- comprimento da sequência: 128 *tokens*;
- temperatura: 0.1;
- número de *tokens* gerados: 3.

Ademais, seguiu-se um protocolo de classificação baseado em duas classes possíveis para a rotulação:

- Classe 0 (zero): sentença sem conteúdo criminoso;
- Classe 1 (um): sentença com indício de envolvimento criminoso.

Prompt

Tendo em vista que os modelos LLM têm seu desempenho também relacionado ao seu *prompt*, foi aplicado o seguinte *input*:

Você é um perito criminal forense especializado em análise de linguagem. Sua tarefa é classificar sentenças de acordo com o seguinte protocolo de classificação de conteúdo com envolvimento criminoso, selecione apenas uma única classe com os números "0", "1", RESPONDA APENAS COM UM NÚMERO:

Protocolo de Classificação:
Classe 0 (zero) - Sentença sem conteúdo criminoso
- Sentença: "Fala mano, blz?"
- Classificação: 0
- Sentença: "Pode pegar o Marquinhos em casa?"
- Classificação: 0
- Sentença: "kkkkkkkkkkkkkkkkkkk"
- Classificação: 0

Classe 1 (um) - Sentença com indício de envolvimento criminoso
- Sentença: "Os cara estam com armas, dando tiro"
- Classificação: 1
- Sentença: "Ei, me vende um bagulho aih!"
- Classificação: 1
- Sentença: "Vc quer fumar uma maconha, vai um baseado aih?"
- Classificação: 1

Considere o contexto a seguir:

[SENTENÇA ANTERIOR] + [SENTENÇA ATUAL] + [SENTENÇA POSTERIOR]

Classificação desta sentença: [SENTENÇA ATUAL]

Esse protocolo foi aplicado aos modelos elencados no Capítulo 4, desses, as melhores respostas, sintaticamente, foram obtidas com o "Llama3 Dolphin Portuguese v0.3" [1] e com o "Llama3 Instruct Portuguese v0.3" [2], uma vez que foram obtidas todas as respostas seguindo o padrão proposto nos exemplos do *prompt*: "Classificação: #", onde é possível uma clara identificação da classe sugerida.

Por outro lado, outros modelos testados apresentaram respostas inconsistentes, não seguindo o padrão solicitado no *prompt*, apesar de terem sido submetidos aos mesmos parâmetros de processamento. Essas inconsistências dificultam a interpretação automática dos resultados e comprometem a confiabilidade da rotulação gerada.

A seguir, são apresentados exemplos de respostas inconsistentes fornecidas por alguns LLMs, evidenciando os desvios em relação ao formato proposto:

- Modelo: Llama 3 [3];
  - "Classificação: _____";
  - "Classificação: _____ (";
  - "Classificação:?????";
- Modelo: Llama3 Dolphin Portuguese v0.4 [4];
  - "Classificação:user";
- Modelo: Llama Tom Cat [5];
  - "Classificação: _____";
  - "Classificação:?????";
  - "Classificação: _____ ("

[1] adalbertojunior/Llama-3-8B-Dolphin-Portuguese-v0.3
[2] adalbertojunior/Llama-3-8B-Instruct-Portuguese-v0.3
[3] meta-llama/Meta-Llama-3-8B
[4] adalbertojunior/Llama-3-8B-Dolphin-Portuguese-v0.4
[5] rhaymison/Llama-3-portuguese-Tom-cat-8b-instruct

Para avaliar o desempenho do modelo, todas essas sentenças, elaboradas a partir das legendas do filme Cidade de Deus, foram classificadas manualmente pelo pesquisador desta dissertação, servindo como referência humana para a análise comparativa.

Posteriormente, os resultados das classificações realizadas pelo modelo foram avaliados utilizando a métrica *F1-Score*. Além disso, foi conduzida uma análise de consistência para verificar o alinhamento entre a classificação gerada pelo modelo com melhor desempenho (em termos de *F1-Score*) e a referência humana.

Os valores resultantes dessa avaliação estão apresentados nas Tabelas 5.3, que destacam os principais resultados para os modelos "Llama3 Dolphin Portuguese v0.3" e "Llama3 Instruct Portuguese v0.3", respectivamente.

Tabela 5.3: Comparativo de *F1-Score* entre modelos

Classe / Média	**Llama3 Dolphin Portuguese v0.3**	**Llama3 Instruct Portuguese v0.3**
Classe 0	0.88	0.68
Classe 1	0.86	0.76
Média Macro	0.87	0.72
Média Ponderada	0.87	0.72

Fonte: Elaborado pelo autor.

Com base nos resultados obtidos, foi possível observar que o modelo "Llama3 Dolphin Portuguese v0.3" apresentou o melhor desempenho no processo de rotulação de dados. No entanto, identificou-se a necessidade de uma análise adicional para verificar a influência do hiperparâmetro de temperatura (hiperparâmetro do LLM) na eficácia do modelo, bem como na consistência de suas classificações.

Para essa análise, a mesma rotulação foi executada mais 4 (quatro) vezes utilizando diferentes valores para o parâmetro de temperatura (*ceteris paribus*): 0.3, 0.5, 0.7 e 0.9. Os resultados obtidos estão apresentados na Tabela 5.4, destacando o impacto desse hiperparâmetro nas métricas de desempenho, particularmente no *F1-Score*. Essa análise objetiva determinar o valor de temperatura mais adequado para balancear a criatividade e a precisão das respostas geradas pelo modelo.

Tabela 5.4: Resultados de *F1-Score* para diferentes valores de temperatura.

Temperatura	***F1-Score***
0.1	0,89
0.3	0,87
0.5	0,89
0.7	0,82
0.9	0,82

Fonte: Elaborado pelo autor.

Como demonstrado nos resultados da Tabela 5.4, os maiores valores de desempenho foram obtidos com as temperaturas 0.1 e 0.5. Para a continuidade do projeto, foi decidido, por discricionariedade, adotar a temperatura de 0.1 como referência. Essa escolha fundamenta-se na tendência de temperaturas mais baixas conduzirem os modelos à geração de sentenças mais repetitivas e claras, característica desejável para o contexto da rotulação, enquanto valores mais altos podem introduzir maior aleatoriedade (NAKAISHI; NISHIKAWA; HUKUSHIMA, 2024).

Com o modelo e seus hiperparâmetros definidos – inclusive a temperatura, o mesmo foi submetido a 4 (quatro) processamentos adicionais, totalizando 5 (cinco) rodadas. Os resultados globais obtidos estão apresentados na Tabela 5.5, destacando o desempenho médio em termos de métricas como *F1-Score* e consistência entre as rodadas.

Tabela 5.5: Resumo Global: 2 classes - 5 rodadas de rotulação

Métrica	**A**	**B**	**C**	**D**	**E**
Precisão	0.87	0.84	0.85	0.85	0.87
Recall	0.87	0.83	0.85	0.85	0.87
F1-Score	0.87	0.83	0.85	0.85	0.87

Fonte: Elaborado pelo autor.

Mesmo apresentando resultados relevantes em seu desempenho, a comparação entre a classificação realizada pelo modelo e a rotulação feita pelo analista humano indicou a necessidade de uma análise mais aprofundada sobre a consistência dos rótulos gerados. Para isso, foi proposta a avaliação da concordância entre os rótulos atribuídos pelo modelo durante os 5 (cinco) processamentos mencionados anteriormente.

Essa análise foi conduzida por meio do teste Fleiss' Kappa, conforme descrito em Moons & Vandervieren (2023), o qual é utilizado para avaliar a concordância entre múltiplos avaliadores ou processamentos. Os resultados desse teste foram interpretados com base nos parâmetros apresentados na Tabela 5.6, permitindo uma avaliação objetiva da consistência dos rótulos designados pelo modelo ao longo das rodadas de processamento.

Tabela 5.6: Fleiss'Kappa: parâmetros de interpretação de resultados

Estatística K	**Nível de Concordância**
<0.00	pobre
0.00-0.20	leve
0.21-0.40	justa
0.41-0.60	moderada
0.61-0.80	substancial
0.81-1.00	quase perfeita

Fonte: Elaborado pelo autor, adaptado de Landis & Koch (1977).

A partir desta avaliação obteve-se um valor de **K = 0.92**, o que demonstra um elevado nível de consistência do modelos durante os 5 processamentos de rotulação de sentenças. Esse resultado reforça a confiabilidade na aplicação da técnica como apoio à criação de uma base de dados rotulada, mesmo considerando o perfil estocástico inerente aos modelos de linguagem.

A fim de elucidar a diferença ocasional apresentada na rotulação, relacionado ao perfil estocástico dos modelos de linguagem, apresentamos na sequência um exemplo de sentença que recebeu 4 (quatro) rotulações "**1**" e na rodada "**B**" a rotulação "**0**", conforme Tabela 5.7.

> *"O que é que ele faz? Não faz nada. - Calma aí. Só porque tu deu a ideia. - Então, porra? Quantas vezes eu vou ter que falar que tu é moleque? Quantas vezes? Tu vai ficar na escolta, morou? Vam'bora. Vai chegar tua hora. Fica tranquilo. Tu 'tá muito afoito pro meu gosto. Aí, piranha, essa porra aqui é um assalto." (MEIRELLES, 2002)*

Ao final, de todo esse processo de teste e seleção, foi escolhido o modelo para ser utilizado na classificação dos dados reais seguindo a seguinte configuração:

Tabela 5.7: Classificação em diferentes rodadas

Rodadas	**A**	**B**	**C**	**D**	**E**
Classificação	1	0	1	1	1

Fonte: Elaborado pelo autor.

- LLM: "Llama3 Dolphin Portuguese v0.3";
- temperatura: 0.1;
- comprimento da sequência: 512 *tokens*;
- tokens gerados: 3.

5.4.2 Rotulação de dados reais com LLM

Uma vez selecionado o modelo e confirmada sua viabilidade com dados simulados, conforme discutido na seção anterior, esta pesquisa passou a aplicar a mesma metodologia utilizando os dados reais dos nove casos analisados.

É importante destacar que a quantidade de *tokens* utilizada nesta fase de rotulação foi de até 512, valor este definido com objetivo de proporcionar um contexto mais amplo para compreensão do modelo e também por ser o limite máximo suportado pelo modelo BERT (DEVLIN et al., 2019), do qual o BERTimbau é derivado.

Assim, as sentenças foram submetidas à rotulação pelo modelo de LLM selecionado e *prompt* conforme Seção 5.4.1, utilizando os seguintes parâmetros:

- comprimento da sequência: 512 *tokens*;
- temperatura: 0.1;
- *tokens* gerados: 3.

5.4.3 Rotulação humana de dados reais

Para viabilizar a avaliação da capacidade dos modelos de processamento de linguagem na tarefa de classificação de sentenças, tornou-se necessária a rotulação manual dos dados reais empregados nesta pesquisa.

Ressalta-se que essa rotulação manual foi realizada pelo pesquisador, atividade que durou em torno de 90 horas de execução junto à sede da PCPR, e considerada, nesta pesquisa, como a referência "real" para fins de avaliação.

Dada a aplicação da técnica *Leave On Group Out* (detalhada na Seção 4.11), é relevante identificar a distribuição da rotulação entre os casos analisados, uma vez que a grande variação entre eles pode impactar diretamente os resultados obtidos nos experimentos.

A Tabela 5.8 apresenta a distribuição dos dados por rótulo para os casos analisados na pesquisa.

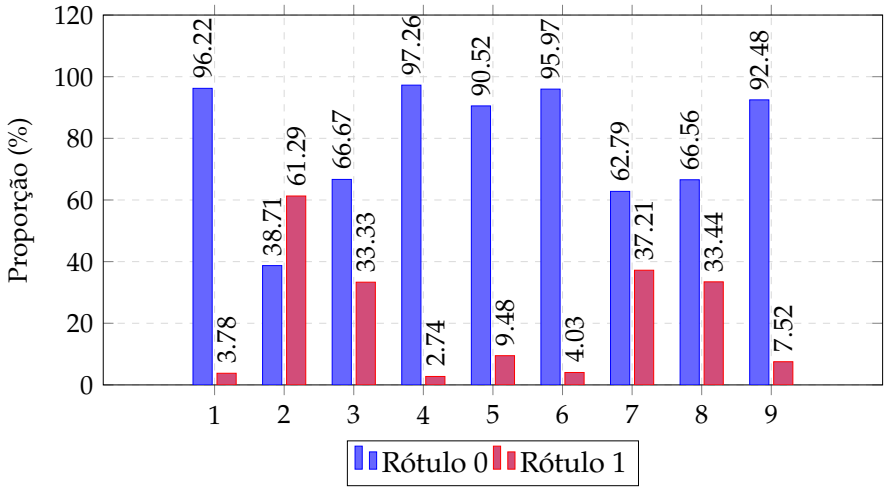
Tabela 5.8: Distribuição de classes dos casos analisados

Caso	**Rótulo: 0**	**Rótulo: 1**	**Total**
1	280	11	291
2	24	38	62
3	6	3	9
4	2.866	81	2.947
5	544	57	601
6	357	15	372
7	54	32	86
8	946	476	1.422
9	1.044	85	1.129
Total	**6.121**	**798**	**6.919**

Fonte: Elaborado pelo autor.

Ao se observar a Tabela 5.8, nota-se uma grande variação no tamanho dos casos, como no caso 3, com apenas 9 sentenças, e no caso 4, com 2.947. Também se observa uma disparidade na distribuição dos rótulos: o caso 4 apresenta apenas 2,7% de rótulos 1, enquanto no caso 2 esse valor chega a 61,3%, conforme Figura 5.5. Por esse motivo a importância da aplicação da técnica de balanceamento de dados, utilizada no intuito de corrigir tais distorções.

Figura 5.5: Distribuição percentual dos rótulos por caso



Fonte: Elaborado pelo autor.

5.5 Variáveis de Treinamento

Tal como discorrido no Capítulo 4, e com o intuito de explorar diferentes formas de extração de conhecimento a partir dos dados obtidos, definiu-se o uso de três variáveis de entrada para treinamento: *sentence*, *keyphrase* e *kp + sent*, descritas a seguir:

- *sentence*: concatenação das mensagens trocadas, limitada a até 512 *tokens*, agrupadas por destinatário ou grupo e ordenadas temporalmente;
- *keyphrase*: palavras-chave (ou frase-chave) extraídas a partir da *sentence*, conforme o rótulo atribuído à sentença:
  - rótulo 1: extração realizada manualmente pelo pesquisador;
  - rótulo 0: extração automática com a biblioteca *KeyBERT* [6].
- *kp + sent*: concatenação de *keyphrase* com *sentence*.

Ressalta-se que a *keyphrase* foi concatenada antes da *sentence* porque a *sentence* já pode atingir o limite de 512 *tokens*. Dessa forma, caso a frase-chave fosse

[6]Extração automática com *KeyBERT* (GROOTENDORST, 2020) utilizando o modelo BERTimbau (*base*), considerando candidatos com um a dois termos (*unigramas* e *bigramas*) e selecionando as cinco frases-chave mais relevantes.

inserida ao final, haveria risco de não ser considerada pelo modelo devido ao truncamento durante o processamento.

5.6 Considerações Finais

Este capítulo apresentou uma visão detalhada da execução dos experimentos, apresentando a infraestrutura de processamento – tanto para dados reais como para simulados – e o conjunto de dados real, composto por nove casos com alta heterogeneidade (variando de 9 a 2.947 sentenças).

A avaliação de desempenho dos modelos transcritores revelou a superioridade dos modelos Whisper sobre o Vosk, levando à seleção do modelo Whisper Small por seu melhor desempenho considerando eficácia (F1-Score médio de 0,69 em dados reais) e eficiência de processamento.

Na fase de rotulação, a aplicação do modelo de linguagem "Llama3 Dolphin Portuguese v0.3" para pré-rotulação demonstrou alta consistência em dados simulados (Fleiss' Kappa 0,92), podendo ser considerado como uma ferramenta viável para a aplicação aqui proposta.

Os resultados deste processo são demonstrados no Capítulo 6, que apresentará e discutirá os resultados obtidos com a validação LOGO para os 30 cenários supervisionados, comparando-os com o método atual utilizado pela PCPR e o desempenho do LLM para a classificação de sentenças.

6

RESULTADOS

Com o objetivo de apresentar os resultados obtidos a partir da aplicação do método proposto e de responder às questões que orientaram o desenvolvimento desta pesquisa, este capítulo detalha a avaliação do desempenho dos modelos e técnicas de Processamento de Linguagem Natural (NLP) empregados na tarefa de classificação de sentenças.

Tal como descrito na metodologia desta pesquisa, os modelos de Processamento de Linguagem Natural (NLP) foram avaliados em termos de desempenho considerando os cenários apresentados na Seção 4.11.

Para isso, adotou-se o método de validação *Leave One Group Out* (LOGO), escolhido por sua capacidade de avaliar a generalização dos modelos em grupos distintos de dados, aproximando-se de situações reais em que o modelo precisa lidar com casos completamente novos.

A aplicação desse protocolo de validação, em conjunto com diferentes arquiteturas de modelos, estratégias de extração de características e técnicas de pré-processamento textual, teve como objetivo responder, de forma empírica e sistemática, às Questões de Pesquisa 1 e 3.

A primeira diz respeito à eficácia das técnicas de classificação de sentenças na identificação de conteúdo potencialmente criminoso, com vistas ao suporte em investigações forenses; já a terceira busca avaliar em que medida o ajuste fino (*fine-tuning*) de modelos de NLP pré-treinados pode contribuir para a ampliação da capacidade de extração de conhecimento a partir de textos em linguagem natural.

Preliminarmente, procedeu-se à avaliação dos cenários propostos - com-

preendendo o balanceamento amostral, o método de representação textual e a variável de treinamento - com o objetivo de aferir o desempenho geral das estratégias de *feature engineering* desenvolvidas, independentemente do classificador utilizado.

Em um segundo momento, os resultados foram analisados a partir da aplicação dos diferentes classificadores em cada cenário, permitindo uma avaliação mais detalhada da interação entre as estratégias de extração de características e os algoritmos de classificação.

6.1 Técnica Atual: Busca de Palavras-chave

A fim de verificar o ganho real proporcionado pelo método proposto neste trabalho, procedeu-se à avaliação da técnica atualmente empregada pela PCPR, baseada na busca de palavras-chave nas mensagens extraídas dos aparelhos periciados.

Para esse isso, utilizou-se a lista oficial de palavras-chave adotada pelos peritos, aplicando-se o procedimento de busca sobre a base de dados previamente estruturada. Dessa forma, as sentenças que continham ao menos uma dessas palavras-chave foram classificadas como "com indício criminoso" (*1*); caso contrário, foram classificadas como "sem indício criminoso" (*0*).

Ao final da análise, obteve-se um valor da média ponderada de *F1-Score* de 0,552 (utilizando mesmo método de cálculo definido na Equação 4.4), como pode ser percebido na Figura 6.1.

Além disso, ao se avaliar o desempenho do método de busca por palavras-chave caso a caso, observa-se uma significativa variação nos valores de *F1-Score*, conforme ilustrado na Figura 6.1. Os resultados variam entre 0,44 e 0,86, sendo este último obtido em um caso com apenas nove amostras, o que limita a confiabilidade estatística do valor alcançado, dada a baixa representatividade do cenário.

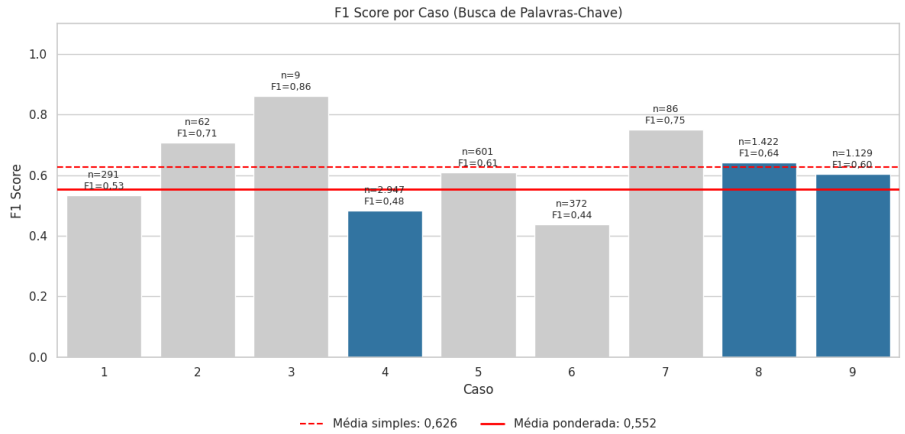
Por outro lado, ao se analisar os casos com mais de mil amostras (colunas em azul), observa-se uma queda representativa no desempenho do método, com valores de *F1-Score* chegando a 0,48 no caso 4, qual possui o maior número de amostras (2.947). Esse desempenho inferior nos cenários mais de maior demanda textual sugere uma baixa capacidade de generalização da técnica de busca por palavras-chave, especialmente quando confrontada com uma maior

diversidade lexical e estrutural dos textos.

Essa discrepância evidencia uma fragilidade do método em contextos mais complexos, nos quais a simples presença de termos-chave não é suficiente para capturar a semântica necessária à correta identificação de sentenças relevantes.

Portanto, ainda que esse método de classificação de sentenças à partir da busca de palavras-chave possa representar uma solução tecnológica viável, sua aplicação em contextos reais e mais amplos pode gerar resultados limitados, o que reforçaria a necessidade de abordagens mais robustas - como o uso de modelos supervisionados baseados em aprendizado de máquina ou de técnicas semânticas baseadas em modelos pré-treinados ou até mesmo LLM - capazes de lidar com a variabilidade linguística presente nos dados reais.

Figura 6.1: Resultados do *F1-Score* por caso com aplicação da técnica de busca de palavras-chave



Fonte: Elaborado pelo autor.

6.2 Avaliação da Rotulação com LLM

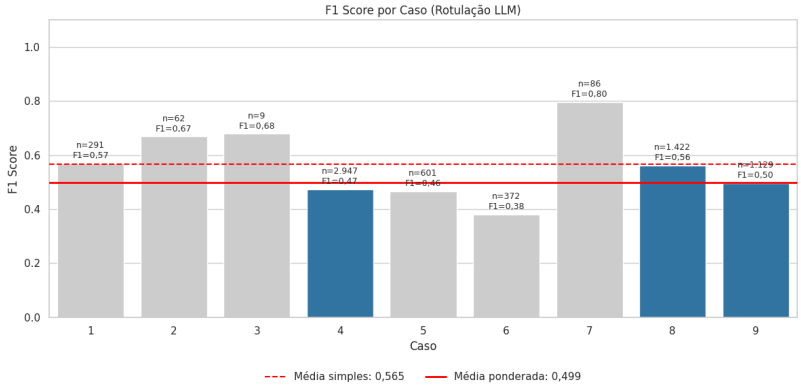
Em termos gerais, observa-se que o desempenho do modelo apresentou certa oscilação entre os casos analisados, com valores de *F1-Score* variando de 0,38 a 0,80, e média simples de 0,565. Esse resultado, entretanto, torna-se menos favorável quando se adota a média ponderada pelo número de amostras de cada caso (Equação 4.4), métrica mais representativa do conjunto avaliado: nesse cenário, o desempenho médio reduz-se para 0,499.

Essa discrepância pode ser explicada, em grande medida, pelo comporta-

mento do modelo nos casos com maior volume de dados. O caso 4, por exemplo, que reúne o maior número de amostras (n=2.947), obteve *F1-Score* de 0,47, contribuindo de forma desproporcional para a queda da média ponderada. Em contraste, o caso 7 alcançou *F1-Score* de 0,80, porém com apenas 86 amostras, exercendo impacto limitado no resultado global.

O padrão observado sugere que o modelo enfrenta maior dificuldade em manter desempenho estável quando exposto a conjuntos extensos e potencialmente mais heterogêneos, nos quais a diversidade lexical, a presença de ruído e a variedade de contextos tendem a ser maiores. Assim, embora o desempenho em casos menores indique capacidade de rotulação adequada em determinados contextos, a análise ponderada evidencia limitações relevantes para a aplicação do método em cenários que concentram a maior parte dos dados.

Figura 6.2: Resultados do *F1-Score* por caso com rotulação feita por modelo LLM



Fonte: Elaborado pelo autor.

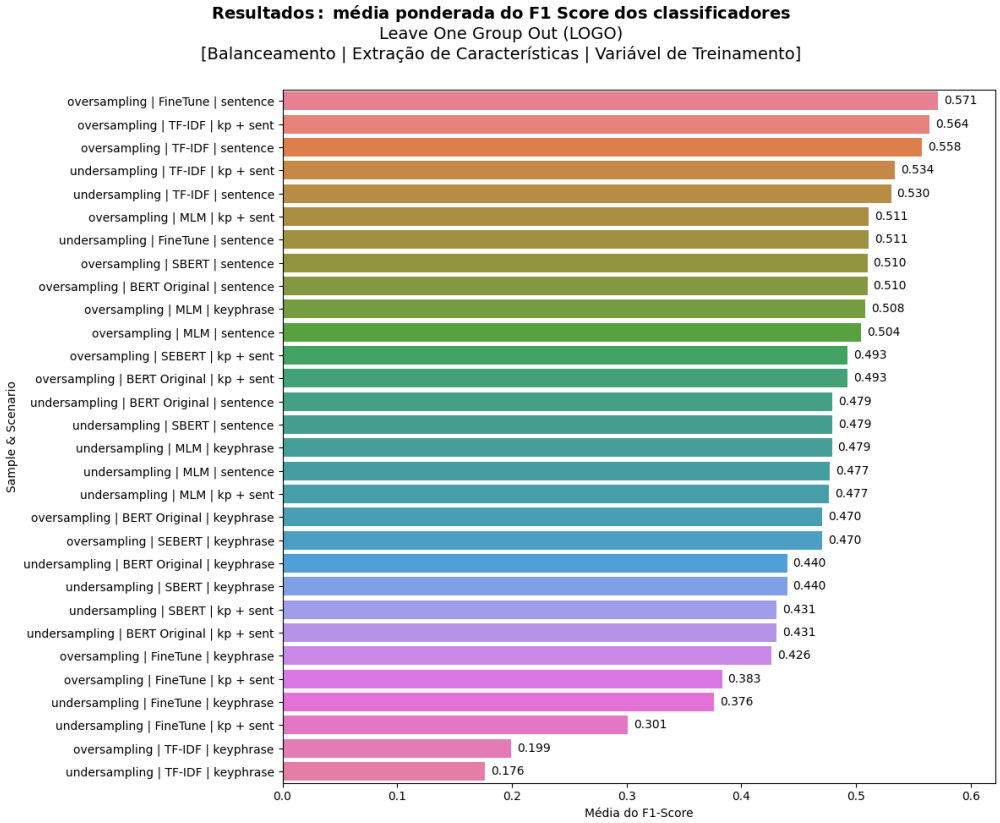
6.3 Avaliação: *Leave One Group Out* (LOGO)

Tal como apresentado na Seção 4.11, os dados foram submetidos à técnica de validação Leave-One-Group-Out (LOGO), com o objetivo de simular uma situação mais realista.

Nessa abordagem, os dados de teste - caso ou grupo - permanecem completamente isolados durante todas as fases do treinamento, incluindo o ajuste fino (quando aplicável) e o treinamento dos classificadores.

Os resultados obtidos a partir dessa modelagem são apresentados na Figura 6.3.

Figura 6.3: Resultados *Leave One Group Out* (LOGO): média ponderada do *F1-Score* para os cenários



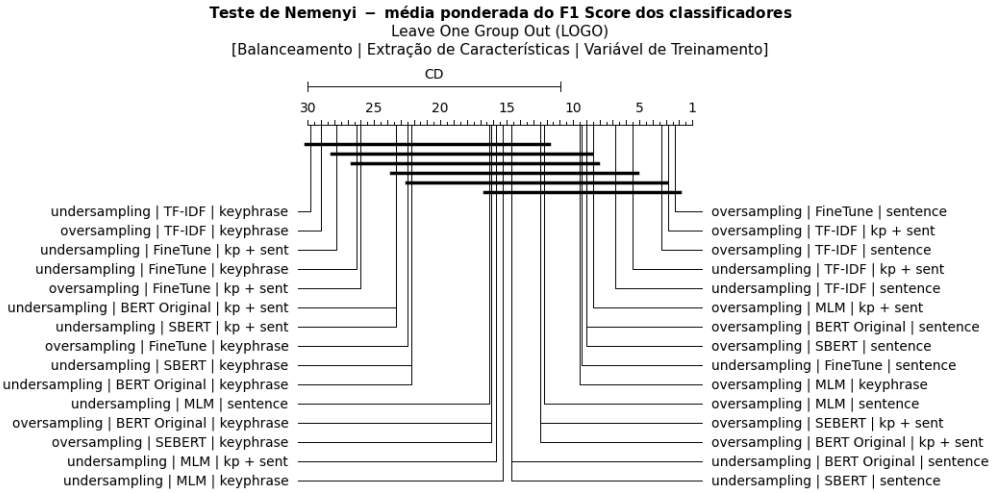
Fonte: Elaborado pelo autor.

Observa-se que, também nesse contexto, o cenário "***Fine-Tune | sentence***" com aplicação de *oversampling* obteve o melhor desempenho médio, atingindo um *F1-Score* de 0,571.

Em seguida, destacam-se quatro cenários baseados no uso de TF-IDF como técnica de extração de características, com pontuações médias entre 0,564 e 0,530, o que evidencia sua competitividade, especialmente quando combinado com estratégias de balanceamento adequadas.

Para verificar a existência de diferenças estatisticamente significativas entre os cenários, aplicou-se o teste de Nemenyi, cujos resultados estão dispostos na Figura 6.4.

Figura 6.4: Teste Nemenyi: *F1-Score* para os cenários



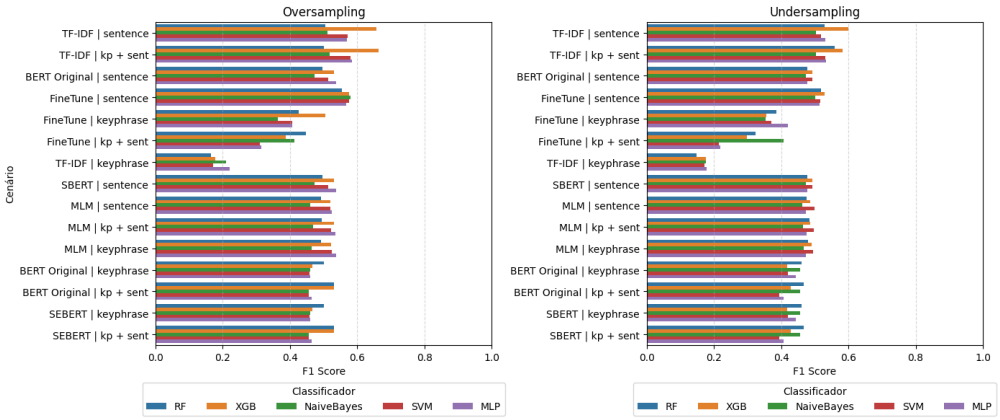
Fonte: Elaborado pelo autor.

Conforme os resultados do teste, é possível afirmar que há diferenças estatísticas significativas entre alguns cenários. No entanto, os cenários com médias de *F1-Score* entre 0,571 ("**oversampling | FineTune | sentence**") e 0,477 ("**undersampling | MLM | sentence**") não apresentaram diferenças significativas entre si, compondo assim um grupo estatisticamente equivalente sob a ótica do teste de Nemenyi.

Quando observado o desempenho dos classificadores, nota-se que, em alguns casos, o extrator de características TF-IDF obteve valores superiores aos demais, sendo o único a ultrapassar consistentemente a marca de 0,60 no *F1-Score* médio em múltiplas combinações. Esse desempenho é particularmente evidente nos cenários que utilizaram as combinações de variáveis *sentence* ou *kp + sent* e aplicaram a técnica de *oversampling*.

Assim como apresentado na Figura 6.5, tais resultados posicionam o TF-IDF como uma alternativa simples e eficaz de extração de características, mesmo quando comparada a métodos baseados em modelos pré-treinados como *FineTune*, *SBERT* ou *MLM*.

Figura 6.5: Resultados *Leave One Group Out* (LOGO): *F1-Score* dos classificadores para os cenários



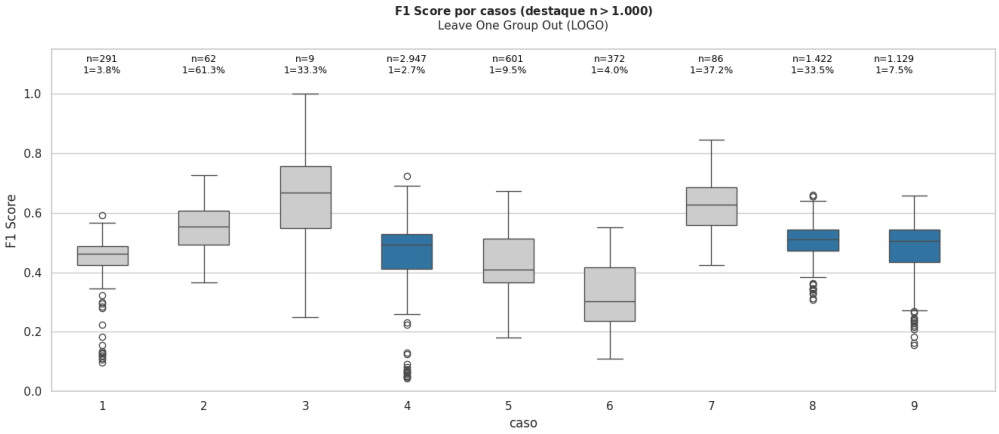
Fonte: Elaborado pelo autor.

Ainda que modelos como o "***FineTune | sentence***" tenham alcançado médias elevadas de desempenho, é relevante destacar que a combinação "***TF-IDF | kp + sent***" com ***oversampling*** também obteve desempenho competitivo e estatisticamente semelhante (conforme o teste de Nemenyi), o que sugere que abordagens tradicionais, quando bem combinadas com técnicas de balanceamento e escolha de classificadores adequados, permanecem viáveis em tarefas complexas de classificação textual, especialmente para o contexto aqui proposto de teste com amostras totalmente isolada do treinamento.

Um aspecto relevante a ser destacado diz respeito à composição dos grupos utilizados na validação LOGO, conforme ilustrado na Figura 6.6. Nota-se, por exemplo, que o grupo 3 conseguiu resultar no desempenho máximo (*F1-Score* = 1,0). Esse resultado, no entanto, deve ser interpretado com cautela, pois o grupo continha apenas nove amostras. Nesse cenário, o modelo foi treinado com uma base muito extensa (excluindo apenas nove sentenças), o que aumenta significativamente a chance de generalização correta sobre esse subconjunto, porém com baixa representatividade estatística.

Por outro lado, o grupo 4, que concentra o maior número de amostras (2.947), obteve uma mediana de desempenho consideravelmente inferior. Isso se justifica pelo fato de que, ao ser reservado como conjunto de teste, esse grupo representou um desafio substancial: além do volume expressivo de amostras a serem corretamente classificadas, essas sentenças não contribuíram para o treinamento do modelo, o que pode ter limitado sua capacidade de generalização.

Figura 6.6: Resultados *Leave One Group Out* (LOGO): *F1-Score* dos classificadores por caso



Fonte: Elaborado pelo autor.

De forma geral, é possível observar que os grupos com maior número de amostras (casos 4, 8 e 9, todos com mais de mil exemplos) tendem a apresentar *F1-Score* medianos em torno de 0,5. Isso reforça a ideia de que o LOGO impõe uma avaliação rigorosa da capacidade preditiva do modelo, uma vez que há maior separação entre os dados de treino e teste, com implicações diretas na variabilidade e robustez do desempenho observado.

6.4 Seleção do modelo de classificação

Assim, como definido na Seção 4.11, a seleção do modelo se deu ao considerar seu respectivo desempenho na validação LOGO, pelo fato desta ter maior proximidade com a atividade pericial quando da necessidade de avaliação do conteúdo de mensagens a partir de um contexto novo e que muitas vezes pode ser totalmente distinto do algum outro já conhecido.

Consequentemente à aplicação desta metodologia, o modelo selecionado foi:

- **oversampling | TF-IDF | kp + sent | XGB**

Tabela 6.1: Ranking dos modelos *F1-Score* - avaliação LOGO

Ordem	Modelo	Média ponderada *F1-Score*
1	`oversampling | TF-IDF | kp + sent | XGB`	0,663
2	`oversampling | TF-IDF | sentence | XGB`	0,656
3	`undersampling | TF-IDF | sentence | XGB`	0,601
4	`oversampling | TF-IDF | kp + sent | MLP`	0,584
5	`undersampling | TF-IDF | kp + sent | XGB`	0,582
6	`oversampling | TF-IDF | kp + sent | SVM`	0,580
7	`oversampling | FineTune | sentence | NaiveBayes`	0,580
8	`oversampling | FineTune | sentence | DecisionTree`	0,576
9	`oversampling | FineTune | sentence | XGB`	0,575
10	`oversampling | FineTune | sentence | SVM`	0,575

Fonte: Elaborado pelo autor.

Ao analisar os resultados obtidos, observa-se que de dez modelos com melhor desempenho, seis deles utilizaram o extrator de características TF-IDF, algo que ao primeiro olhar pode até parecer um resultado surpreendente, por se tratar de um algorítimo de menor complexidade e necessidade de processamento, como os modelos pré-treinados.

Situação semelhante já fora observada em outros domínios de pesquisa. Por exemplo, Kılıç (2025) comparou o desempenho de modelos econométricos — como Autoregressive Fractionally Integrated Moving Average (ARFIMA) e Heterogeneous Autoregressive Model (HAR) — com diferentes abordagens de *Machine Learning* (ML), incluindo Extreme Gradient Boosting, redes neurais *feedforward* profundas e redes recorrentes (BRNN, LSTM, LSTM-A e GRU), aplicados à previsão de volatilidade realizada do índice S&P 500. Nesse estudo, os modelos econométricos apresentaram desempenho superior às técnicas de aprendizagem de máquina, sendo ainda considerados mais simples e interpretáveis para a tarefa analisada.

Assim, o desempenho verificado em nossos experimentos se justifica ao considerar as características específicas dos dados analisados. Os casos periciais utilizados apresentam contextos altamente diversos, com variações significativas no estilo de linguagem, contexto social, perfil dos interlocutores e formas de envolvimento criminoso, além de um vocabulário de certa maneira limitado e recorrente ao tratar do tema em pesquisa.

Essa diversidade impacta diretamente o desempenho de modelos que dependem de aprendizado contextual, como os ajustados via *Fine-Tuning*, pois a vali-

dação do tipo LOGO implica sempre a avaliação sobre um caso completamente inédito — cujo padrão linguístico não foi observado durante o treinamento. Nessa configuração, mesmo modelos ajustados tendem a perder desempenho por não reconhecerem as particularidades do novo caso.

Nesse cenário, o TF-IDF demonstrou maior capacidade de adaptação ao tipo de conteúdo predominante nas mensagens de *WhatsApp*, baseando sua representação exclusivamente na frequência dos termos, sem depender de estruturas semânticas complexas. Essa abordagem se mostrou especialmente robusta diante de textos marcados por informalidade, uso de gírias, neologismos, erros ortográficos e abreviações, aspectos que frequentemente reduzem a eficácia de representações mais sofisticadas. Assim, embora mais simples, o TF-IDF apresentou, neste estudo, desempenho superior em termos de generalização, revelando-se a alternativa eficiente, dentre as aqui avaliadas, para contextos forenses com alta variabilidade textual.

Além disso, modelos pré-treinados como BERT ou LLM são, em geral, treinados em corpora amplos e de linguagem formal, o que pode reduzir sua eficácia na interpretação de textos altamente informais ou com grafias não padronizadas, assim como aponta Zhao et al. (2023) sobre a necessidade de ajuste fino desses modelos para sua adaptação à tarefas específicas. Nessas condições, a ausência de adaptação suficiente por meio de *fine-tuning* pode ter comprometido a capacidade desses modelos de capturar nuances semânticas específicas, ao passo que o TF-IDF, por depender exclusivamente da frequência e distribuição dos termos no próprio corpus, adapta-se de forma direta à realidade linguística da base.

Dessa forma, o desempenho observado corrobora achados de estudos prévios que indicam que a escolha da técnica de representação textual deve considerar não apenas o estado da arte, mas também a compatibilidade entre o modelo e a natureza do dado analisado. Conforme observado por Minaee et al. (2020), modelos com arquiteturas mais simples podem superar abordagens mais complexas quando aplicados a bases com elevada variabilidade sintática, justamente por se adaptarem de forma mais direta à distribuição e à natureza dos dados.

Outro ponto a ser destacado é que o modelo com maior desempenho foi treinado com dados balanceados por meio da técnica de *Random Oversampling* (ROS), que consiste em ampliar a quantidade de amostras da classe minoritária até igualá-la à da classe majoritária.

Essa abordagem, apesar de aumentar a demanda computacional, especi-

almente considerando dados deveras desbalanceados como os utilizados na experimentação – aproximadamente uma razão de 1 para 10 entre as classes, resulta em um ganho informacional significativo, visto que a mesma modelagem conseguiu melhorar seu desempenho apenas com a mudança de balanceamento de RUS para ROS, com aumento de 14% comparando os itens 5 e 1 da Tabela 6.1. Tal benefício é particularmente relevante no contexto desta pesquisa, cuja base apresenta uma linguagem específica e que acaba envolvendo apenas um grupo restrito da sociedade.

Quanto às variáveis de entrada utilizadas nos modelos, observa-se que, entre os dez melhores desempenhos apresentados na Tabela 6.1, apenas as representações "**sentence**" e "**kp + sentence**" estiveram presentes, indicando que o uso exclusivo de ***keyphrases*** não foi eficaz para esta tarefa. Além disso, ao comparar os pares de modelos com a mesma configuração, é possível notar que, em alguns casos, a concatenação da ***keyphrase*** à sentença trouxe um leve ganho de desempenho — como no caso dos itens 1 e 2 da mesma Tabela, com incremento de cerca de 0,7%. Por outro lado, também se observou uma ligeira queda (cerca de -1,9%) em modelos como os itens 3 e 5. Assim, não é possível afirmar, com base nesses dados, que a concatenação da ***keyphrase*** à sentença resulta, de forma consistente, em melhor desempenho dos modelos.

Por fim, os testes LOGO evidenciaram como a capacidade de generalização dos modelos varia conforme o caso (grupo) de teste: situações com menor número de amostras ou conteúdo mais homogêneo tendem a inflar o *F1-Score*, enquanto casos maiores e mais diversos desafiam a robustez do modelo de forma mais realista, conforme Figura 6.6.

Diante da avaliação dos resultados obtidos com os modelos aplicados aos diferentes cenários experimentais, torna-se pertinente comparar o desempenho das três abordagens analisadas nesta pesquisa: (i) a classificação com base nos cenários supervisionados desenvolvidos, (ii) a pré-classificação utilizando modelos de linguagem de grande porte (LLM), conforme discutido na Seção 5.4.1, e (iii) o método tradicional de busca por palavras-chave, adotado pela PCPR. A Tabela 6.2 apresenta os valores médios de *F1-Score* obtidos por cada uma dessas abordagens.

Tabela 6.2: Comparação do desempenho médio (*F1-Score*) entre as abordagens avaliadas

Abordagem	***F1-Score* média ponderada dos casos**
Modelo selecionado (cenários supervisionados)	0,663
Busca por palavras-chave (PCPR)	0,552
Modelos de linguagem (LLM)	0,499

Fonte: Elaborado pelo autor.

A comparação entre os valores obtidos permite algumas considerações relevantes. Em primeiro lugar, observa-se que os modelo selecionado apresenta desempenho significativamente superior, em termos relativos, ao método atualmente utilizado pela equipe forense, baseado em busca por palavras-chave. O ganho médio de 0,111 pontos no *F1-Score* representa uma melhoria de 20,11% na capacidade de identificação de sentenças com indícios de envolvimento criminoso. Essa diferença evidencia uma possível limitação da abordagem baseada apenas em vocabulários previamente definidos e reforça o potencial de generalização dos modelos supervisionados. Ainda, vale ressaltar que o modelo selecionado — que utiliza a representação TF-IDF da sentença como entrada em um classificador *XGBoost* — é de baixa complexidade computacional, o que o torna especialmente adequado para cenários com infraestrutura limitada.

Outro ponto importante diz respeito à classificação automática utilizando modelos de linguagem de grande porte (LLM), a qual apresentou o menor desempenho médio entre as abordagens comparadas (*F1-Score* médio de 0,499). Além do baixo desempenho, essa abordagem demanda infraestrutura computacional consideravelmente mais robusta. Como discutido anteriormente nesta pesquisa, a aplicação prática de LLMs exigiu inclusive a quantização do modelo para que fosse possível sua execução local, o que, ainda assim, não resultou em ganhos expressivos de desempenho.

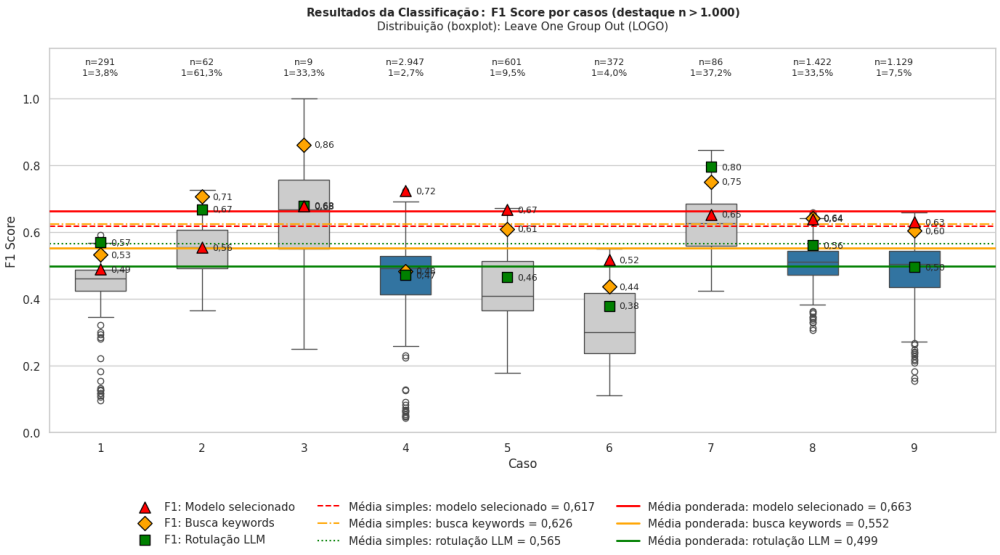
A análise dos resultados por caso, conforme ilustrado na Figura 6.7, reforça a superioridade do modelo supervisionado selecionado (cenário: ***oversampling | TF-IDF | kp + sent | XGB***), o qual obteve o melhor desempenho na maioria dos casos. Em cinco dos nove casos avaliados, o modelo supervisionado alcançou *F1-Score* superiores ou equivalentes aos demais métodos, superando as abordagens alternativas em diversos contextos. Vê-se que nos casos com maior

amostragem (destaque em azul na figura) o modelo supervisionado obteve os melhores resultados.

Especial atenção deve ser dada ao caso 4 — o maior conjunto avaliado, com quase 3.000 amostras. Nesse cenário, o modelo supervisionado alcançou um *F1-Score* de 0,72, enquanto os demais métodos apresentaram desempenhos bastante inferiores (0,48 para palavras-chave e 0,47 para LLM). Esse resultado é particularmente significativo, pois evidencia a robustez do modelo supervisionado em situações que exigem maior capacidade de generalização, seja pela proporção desbalanceada entre rótulos (apenas 2,7% de sentenças positivas), seja pela menor quantidade de conhecimento transferido aos modelos juntamente com uma maior exigência de acerto durante as validações.

Dessa forma, conclui-se que o uso de modelos supervisionados representa uma proposta promissora para o aprimoramento da triagem automatizada de mensagens, sendo uma alternativa viável tanto para substituição quanto para reforço do método atualmente adotado. A combinação de maior eficácia com baixa exigência de recursos computacionais torna essa abordagem especialmente atrativa para instituições com limitações de infraestrutura computacional.

Figura 6.7: Resultados do *F1-Score* para os diferentes métodos de classificação de sentenças



Fonte: Elaborado pelo autor.

6.5 Avaliação com Otimização de Parâmetros - HPO

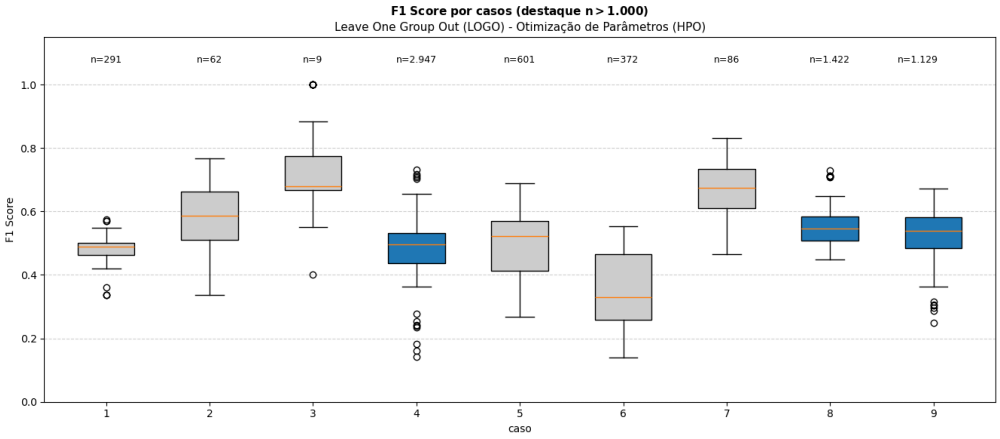
Após a obtenção dos resultados apresentados nas Seções 6.3 e 4.11, aplicou-se a técnica de *Hyperparameter Optimization* (HPO) com o objetivo de explorar, de forma mais sistemática, o potencial do modelo pré-treinado e dos dados empregados no treinamento, verificando se a otimização dos hiperparâmetros seria capaz de produzir desempenho superior ao do modelo previamente selecionado.

Diferentemente da configuração *default* adotada pelas bibliotecas, a *Hyperparameter Optimization* (HPO) conduz uma busca orientada no espaço de hiperparâmetros, avaliando múltiplas combinações (conforme descrito na Seção 4.9) e selecionando aquelas que maximizam o desempenho no conjunto de validação definido pelo protocolo experimental. Dessa forma, pretende-se reduzir o risco de resultados subótimos decorrentes de escolhas arbitrárias de parâmetros como *learning rate*, *epochs* e *batch size*, especialmente em cenários de *fine-tuning* em que tais decisões impactam diretamente a estabilidade do treinamento e a capacidade de generalização.

Assim, a partir deste processamento foi obtida a distribuição dos valores de *F1-Score*, especificamente para cada grupo – conforme Figura 6.8, após a aplicação da técnica de *Hyperparameter Optimization* (HPO) no *fine-tuning*:

Como resultado, foi obtida a distribuição dos valores de *F1-Score* por grupo, apresentada na Figura 6.8. Ao comparar as Figuras 6.6 e 6.8, observa-se a preservação do mesmo padrão global de dispersão: grupos com menor número de amostras tendem a apresentar maiores valores, enquanto grupos com volume superior a mil amostras - destacados em azul - mantêm medianas próximas de 0,50. Esse comportamento sugere que, embora a *Hyperparameter Optimization* (HPO) eleve o desempenho médio em determinados cenários, parte relevante da limitação permanece associada à heterogeneidade e complexidade intrínseca dos grupos mais representativos, nos quais a diversidade linguística e temática tende a ser maior e mais difícil de capturar por modelos ajustados apenas via texto.

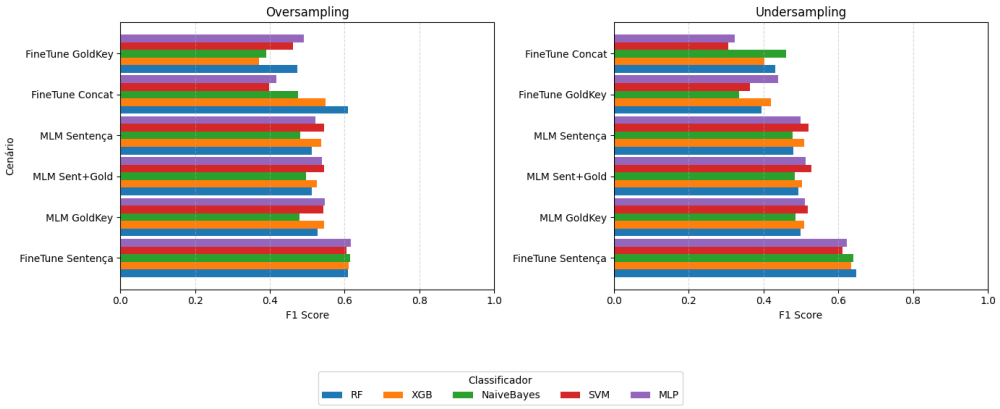
Figura 6.8: Resultados *Leave One Group Out* (LOGO): *F1-Score* dos classificadores por caso, com aplicação do *Hyperparameter Optimization* (HPO)



Fonte: Elaborado pelo autor.

Ao analisar os resultados por cenário e compará-los aos obtidos sem *Hyperparameter Optimization* (HPO), verifica-se melhora consistente, em especial para o cenário "*FineTune* | *sentence*", que passou a apresentar valores acima de 0,60 em múltiplas combinações de balanceamento e classificador, conforme a Figura 6.9 e a Tabela 6.3. Um ponto relevante é que os dez melhores resultados do ranking pertencem ao mesmo cenário ("*FineTune* | *sentence*") — variando apenas entre RUS e ROS e o classificador empregado — o que indica que a otimização favoreceu fortemente esse arranjo específico de representação textual e ajuste do modelo.

Figura 6.9: Resultados *Leave One Group Out* (LOGO): *F1-Score* dos classificadores para os cenários, com aplicação do *Hyperparameter Optimization* (HPO)



Fonte: Elaborado pelo autor.

Tabela 6.3: Ranking dos modelos *F1-Score* - avaliação LOGO, com aplicação do *Hyperparameter Optimization* (HPO)

Ordem	**Modelo**	**Média ponderada *F1-Score***
1	`undersampling	FineTune	sentence	RandomForest`	0,647
2	`undersampling	FineTune	sentence	NaiveBayes`	0,640
3	`undersampling	FineTune	sentence	XGB`	0,634
4	`undersampling	FineTune	sentence	MLP`	0,623
5	`oversampling	FineTune	sentence	DecisionTree`	0,622
6	`oversampling	FineTune	sentence	MLP`	0,616
7	`oversampling	FineTune	sentence	NaiveBayes`	0,615
8	`undersampling	FineTune	sentence	SVM`	0,611
9	`oversampling	FineTune	sentence	XGB`	0,611
10	`oversampling	FineTune	sentence	RandomForest`	0,609

Fonte: Elaborado pelo autor.

O melhor desempenho observado foi de *F1-Score* igual a 0,647, obtido pelo modelo "*undersampling | FineTune | sentence | RandomForest*". Em relação ao melhor valor reportado anteriormente para esse cenário sem otimização (*F1-Score* = 0,518), trata-se de um ganho expressivo, especificamente para este modelo em foco, da ordem de aproximadamente 25%, evidenciando que a escolha de hiperparâmetros tinha papel determinante na performance do *fine-tuning*. Ainda assim, mesmo com a melhora, o resultado otimizado não superou o modelo previamente selecionado ("*oversampling | TF-IDF | kp + sent | XGB*"), que alcançou

F1-Score de 0,663.

Assim, é possível concluir que a técnica de otimização trouxe uma melhora significativa no desempenho dos modelos, todavia não o suficiente para superar o resultado previamente obtido, demonstrando a superioridade arquitetura do *Term-Frequency Inverse Document Frequency* (TF-IDF) para este caso concreto.

Dessa forma, conclui-se que a *Hyperparameter Optimization* (HPO) contribuiu para elevar substancialmente o desempenho dos modelos baseados em *fine-tuning*, tornando-os mais competitivos e reduzindo a distância em relação ao melhor arranjo global. Contudo, os resultados indicam que, para este conjunto de dados e sob o protocolo LOGO, a combinação baseada em *Term-Frequency Inverse Document Frequency* (TF-IDF) com *kp + sent* permaneceu superior. Isso sugere que, no caso concreto, a representação estatística e a modelagem associada ao XGBoost capturam padrões mais estáveis e generalizáveis do que as representações obtidas via ajuste fino, mesmo quando este é cuidadosamente otimizado.

6.6 Avaliação da técnica de *fine-tuning*

Com o intuito de responder a uma das questões propostas no início deste trabalho, avaliou-se se a aplicação da técnica de ajuste fino contribuiu para a melhoria do desempenho do modelo pré-treinado na tarefa de classificação de sentenças.

Os resultados indicaram efeitos distintos[1], variando de acordo com a variável de entrada utilizada. Embora não tenha havido ganhos expressivos de maneira geral, o cenário "**oversampling | FineTune | sentence**" obteve um incremento de 12,07% no *F1-Score* em comparação com o modelo base "**oversampling | BERT Original | sentence**".

De forma semelhante, no cenário com ***undersampling***, esse mesmo par apresentou uma melhora de 6,58%. Esses resultados sugerem que o ajuste fino pode contribuir positivamente quando aplicado a representações textuais mais ricas, como sentenças completas.

Por outro lado, seu impacto mostrou-se negativo em configurações com entradas mais sintéticas, como ***keyphrases*** ou concatenações (***kp + sent***), onde

[1]Considerando a média das médias ponderadas dos modelos (Equação 4.4).

observaram-se quedas de desempenho superiores a 20% em alguns casos.

Isso indica que o benefício do *fine-tuning* depende fortemente da qualidade e expressividade das entradas textuais utilizadas no treinamento, conforme demonstrado nas Tabelas 6.4 e 6.5, com os resultados obtidos na validação LOGO.

Tabela 6.4: Desempenho comparativo entre BERT Original e Fine-Tune com **Oversampling**

BERT Original		**Fine-Tune**		**% *F1-Score***
Variável de treino	***F1-Score***	**Variável de treino**	***F1-Score***	
keyphrase	0,470	keyphrase	0,426	–9,35%
kp + sent	0,493	kp + sent	0,383	–22,19%
sentence	0,510	sentence	0,571	+12,07%

Fonte: Elaborado pelo autor.

Tabela 6.5: Desempenho comparativo entre BERT Original e Fine-Tune com **Undersampling**

BERT Original		**Fine-Tune**		**% *F1-Score***
Variável de treino	***F1-Score***	**Variável de treino**	***F1-Score***	
keyphrase	0,440	keyphrase	0,376	–14,53%
kp + sent	0,431	kp + sent	0,301	–30,17%
sentence	0,479	sentence	0,511	+6,58%

Fonte: Elaborado pelo autor.

6.7 Considerações Finais

Este capítulo apresentou a avaliação sistemática dos modelos, respondendo às questões de pesquisa propostas.

Foi constatado que a técnica de classificação de sentenças, baseada em modelos supervisionados, superou o método atual da PCPR (busca por palavras-chave, *F1-Score* 0,552) e a classificação direta por LLM (*F1-Score 0,499*).

O modelo de melhor desempenho, selecionado pela validação *Leave One Group Out* (LOGO), foi a combinação "oversampling | TF-IDF | kp + sent | XGBoost", que atingiu uma média ponderada de *F1-Score* de 0,663. Este resultado

representou um aumento de 20,11% na eficácia em relação à técnica de palavras-chave, destacando a robustez de uma abordagem mais simples (TF-IDF) em contextos de alta variabilidade e linguagem informal, como nos casos analisados nesta pesquisa.

Como etapa complementar, aplicou-se *Hyperparameter Optimization* (HPO) com o objetivo de investigar se a otimização de hiperparâmetros poderia ampliar o desempenho dos modelos baseados em *fine-tuning*. Os resultados mostraram ganhos substanciais no cenário "*FineTune* | *sentence*", cujas melhores combinações passaram a superar 0,60, atingindo *F1-Score* máximo de 0,647 com "*undersampling* | *FineTune* | *sentence* | *RandomForest*". Embora esse avanço represente um incremento expressivo em relação ao mesmo cenário sem otimização (*F1-Score* = 0,518), ele não foi suficiente para superar o melhor resultado global (0,663) obtido com *Term-Frequency Inverse Document Frequency* (TF-IDF).

7

CONCLUSÃO

As aplicações de diferentes técnicas de inteligência artificial em soluções que auxiliem em atividades profissionais, em diversas áreas de conhecimento, têm se tornado cada vez frequente, especialmente com o recente desenvolvimento exponencial na área de Processamento de Linguagem Natural (NLP), com utilização de modelos pré-treinados capazes de compreender, gerar e fazer inferências sobre a linguagem humana.

Nesse sentido, o presente trabalho focou na geração de um modelo capaz de auxiliar as atividades de perícia criminal em sua análise de mensagens de texto e áudio trocadas em aplicativos de celular, a fim de identificar possíveis indícios de envolvimento criminal.

Para tal, o projeto SICRET II – incentivado pelo Programa de Cooperação Acadêmica em Segurança Públicae Ciências Forenses da CAPES (BRASIL, 2020) – contou com o apoio e parceria da Polícia Civil do Estado do Paraná (PCPR), órgão do Estado do Paraná competente para realização de perícias criminais, o qual disponibilizou casos, amostras periciadas para servirem de base aos processos computacionais. Processos esses que foram executados dentro do ambiente físico e lógico da polícia – *on premise* – dado seu caráter confidencial.

Um dos grandes desafios deste projeto, especialmente no que se refere à criação de modelos de classificação, foi a dependência de uma base de dados rotulada.

Diante dessa necessidade, elaborou-se uma base composta por 9 casos (aparelhos celulares periciados), totalizando 392.416 sentenças e 2.961.033 palavras.

Esse conjunto incluiu tanto os textos trocados em mensagens quanto 25.768

áudios transcritos com o apoio de modelos de transcrição automática.

Na sequência, procedeu-se à rotulação da base de dados, atividade que demandou aproximadamente 90 horas, a fim de viabilizar sua utilização nos experimentos, tanto para o ajuste do modelo BERTimbau quanto para os classificadores tradicionais e grandes modelos de linguagem LLM. Com esses dados, modelos de LLM foram avaliados a fim de verificar sua capacidade de classificação de sentenças, com a identificação de conteúdo relacionado ao tráfico de drogas, tendo sido observado um *F1-Score* de 0,499, média ponderada dentre os casos avaliados, como melhor resultado obtido.

Também foram avaliados 30 cenários experimentais, resultantes da combinação entre o tipo de balanceamento, a forma de representação e a variável de treinamento, aplicados a cinco classificadores distintos, totalizando 150 configurações avaliadas.

Os modelos foram avaliados exclusivamente por meio da validação *Leave One Group Out* (LOGO), adotada como método principal de aferição de desempenho. Essa escolha fundamenta-se na sua capacidade de reproduzir de forma mais fiel a prática forense, na qual é necessário lidar com conteúdos inéditos, em contextos distintos e com estilos de linguagem variados em cada caso analisado.

Ao final, identificou-se o modelo "**oversampling | TF-IDF | kp + sent | XGB**" como o de melhor desempenho – dentre os cenários propostos, alcançando uma média ponderada de *F1-Score* de 0,663, valor significativamente superior (aumento de 32,87%) ao resultado obtido com a utilização do modelo LLM (*F1-Score* = 0,499) e também maior à técnica atualmente empregada pela PCPR baseada em busca de palavras-chave, que apresentou uma média do *F1-Score* de 0,552 para os mesmos casos analisados.

Com relação ao modelo selecionado, convém ressaltar que, devido sua estrutura estar baseada no *Term-Frequency Inverse Document Frequency* (TF-IDF), há uma menor exigência de infraestrutura computacional para seu processamento local, bem como para o seu tempo de processamento, diferente do uso de outros modelos como os pré-treinados e especialmente os baseados em grandes modelos de linguagem que demandam de elevados recursos computacionais e o tempo necessário para a avaliação de todas as sentenças em cada caso, o que pode inviabilizar sua adoção no contexto específico considerado neste trabalho, além de outras questões já discutidas neste trabalho como a inconsistência desses modelos.

Com isso, é importante destacar que a aplicação dos modelos selecionado

resultou em um aumento de 20,11% no *F1-Score* em comparação com o *status quo*, o que representa um avanço expressivo, especialmente considerando a complexidade da tarefa. A identificação de mensagens com indícios de atividade criminosa envolve inúmeros desafios linguísticos, como o uso de gírias, linguagem cifrada, regionalismos, erros ortográficos, variações socioculturais e a diversidade de papéis assumidos pelos interlocutores no contexto da prática criminosa. Nesse cenário, a maior capacidade de generalização e compreensão semântica dos modelos de linguagem pode representar um diferencial importante na triagem automatizada de conteúdo sensível.

Ao considerar as questões iniciais que motivaram esta pesquisa, é possível concluir que:

1. As técnicas de classificação de sentenças com modelos supervisionados mostraram-se eficazes na identificação de indícios de atividade criminosa, com significativa superioridade em relação ao método de busca de palavras-chave, utilizado atualmente pela equipe forense;

2. Os modelos Grandes Modelos de Linguagem (LLM) apresentaram desempenho inferior tanto à busca por palavras-chave — atualmente utilizada pela PCPR — quanto aos modelos classificadores, além de demandarem maior quantidade de recursos computacionais para sua execução;

3. A aplicação da técnica de *fine-tuning* em modelos pré-treinados apresentou resultados mistos, sugerindo que sua eficácia depende diretamente da qualidade e da expressividade das variáveis de entrada utilizadas no treinamento.

Como trabalhos futuros, recomenda-se a ampliação da base de casos avaliados, de forma a garantir maior robustez estatística e representatividade em relação à diversidade de contextos periciais enfrentados pela PCPR. Além disso, propõe-se o aprofundamento na avaliação de novos modelos baseados em LLM, com foco especial em versões mais recentes e otimizadas, ou variantes adaptadas para execução local com menor custo computacional.

Destaca-se também a relevância do uso de técnicas de engenharia de *prompts*, com o objetivo de melhorar a assertividade e a estabilidade das respostas dos modelos de linguagem em tarefas de classificação sensível. Neste sentido, sugere-se ainda a investigação da aplicação de arquiteturas baseadas em *agents*, que permitam a orquestração inteligente de fluxos de análise, triagem e decisão sobre

grandes volumes de mensagens, combinando regras, classificadores tradicionais e modelos generativos.

Além disso, sugere-se a investigação do uso de dados multimodais, incluindo imagens relacionadas aos casos periciados, como fotografias, capturas de tela e outras evidências visuais, que possam ser analisadas isoladamente ou em conjunto com o texto, ampliando o escopo e a precisão das inferências realizadas por modelos híbridos de linguagem e visão computacional.

Por fim, convém avaliar a possibilidade de validação empírica do modelo com o envolvimento de múltiplos peritos criminais, permitindo avaliar a concordância entre os rotuladores e o potencial do sistema como ferramenta complementar de análise.

Este estudo não tem a pretensão de oferecer uma solução definitiva para o problema investigado, sobretudo diante do contínuo avanço das técnicas de Processamento de Linguagem Natural (NLP) e da crescente capacidade computacional disponível.

Ainda assim, esta pesquisa apresenta contribuições relevantes: no campo acadêmico, ao comparar e validar diferentes combinações de modelos e técnicas de análise textual; e, na prática pericial da Polícia Civil do Estado do Paraná (PCPR), ao evidenciar a viabilidade de um modelo capaz de apoiar investigações baseadas em grandes volumes de mensagens, configurando-se como um recurso potencial de suporte à atividade forense.

Referências

Agência Estadual de Notícias. *Seção da Polícia Científica bate recorde de perícias em vestígios cibernéticos em 2024*. Agência Estadual de Notícias do Paraná, 2024. Accessed: 2024-06-21. Disponível em: <https://www.aen.pr.gov.br/Noticia/Secao-da-Policia-Cientifica-bate-recorde-de-pericias-em-vestigios-cibernetico s-em-2024>.

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. [S.l.: s.n.], 2019. p. 2623–2631.

ALVES, J. H. et al. Detecting relevant information in high- volume chat logs: Keyphrase extraction for grooming and drug dealing forensic analysis. In: *2023 International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2023. Disponível em: <http://dx.doi.org/10.1109/ICMLA58977.2023.00299>.

ANATEL. *Evolução dos acessos/densidade de Telefonia Móvel > Acessos Banda Larga Móvel [Brasil, Paraná]*. 2023. <https://informacoes.anatel.gov.br/paineis/aces sos/telefonia-movel>. Acesso em: 23/08/2023 – 23:49h.

ASSEMBLEIA LEGISLATIVA DO ESTADO DO PARANÁ. *Núcleo de Combate aos Cibercrimes elucidou 93% das ocorrências atendidas pelo órgão*. 2023. <http://ww w.assembleia.pr.leg.br/comunicacao/noticias/nucleo-de-combate-aos-cib ercrimes-elucidou-93-das-ocorrencias-atendidas-pelo-orgao>. Acesso em: 21/08/2023 – 22:18h.

BARCELOS, M. Q. et al. O impacto do uso da internet no rendimento acadêmico dos alunos do curso de ciências contábeis de uma i.e.s. privada do interior de minas gerais. *Revista de Administração, Contabilidade e Gestão de Instituições Financeiras*, v. 7, n. 31, p. 122–133, 2019. Disponível em: <https://revistas.fucam p.edu.br/index.php/ragc/article/view/1893/1216>.

BENGIO, Y. et al. A neural probabilistic language model. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 1137–1155, mar. 2003. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=944919.944966>.

BILAL, M.; ALMAZROI, A. A. Effectiveness of fine-tuned bert model in classification of helpful and unhelpful online customer reviews. *Electronic Commerce*

Research, v. 23, n. 4, p. 2737–2757, 2023. Disponível em: <https://doi.org/10.1007/s10660-022-09560-w>.

BRASIL. *Edital nº 16/2020: Programa de Cooperação Acadêmica em Segurança Pública e Ciências Forenses*. 2020. <https://www.gov.br/capes/pt-br/centrais-de-conteudo/01092020_EDITAL_162020.pdf>. Acesso em: 26 dez. 2024.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1010933404324>.

BRITO, L. A. V. *Modelo de classificação multivariável para identificação de enchentes: um estudo empírico no sistema de monitoramento de rios e-noe*. 70 p. Dissertação (Dissertação (Mestrado)) — Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2019.

CAMPOS, R. et al. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, v. 509, p. 257–289, 2020. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025519308588>.

CARNAZ, G. et al. An Annotated Corpus of Crime-Related Portuguese Documents for NLP and Machine Learning Processing. *Data*, MDPI, v. 6, n. 7, p. 71, jun 2021. Disponível em: <https://doi.org/10.3390/data6070071>.

CHEN, G. et al. Small-footprint keyword spotting using deep neural networks. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2014. p. 4087–4091.

Clube de Poesia. *50 Poetas do Clube de Poesia: 1945—1995*. São Paulo: Editora Giordano, 1995. Impresso por Edições Loyola. Publicado em fevereiro de 1995 em comemoração ao cinquentenário do Clube de Poesia, com o apoio do Unibanco.

DEMANT, J. et al. Drug dealing on facebook, snapchat and instagram: A qualitative analysis of novel drug markets in the nordic countries. *Drug and Alcohol Review*, Wiley, v. 38, n. 4, p. 377–385, May 2019. Epub 2019 May 3.

DETTMERS, T.; ZETTLEMOYER, L. *The case for 4-bit precision: k-bit Inference Scaling Laws*. 2023. Disponível em: <https://arxiv.org/abs/2212.09720>.

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

ERRA, U. et al. Approximate tf–idf based on topic extraction from massive message stream using the gpu. *Information Sciences*, v. 292, p. 143–161, 2015.

ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025514008676>.

GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998.

GROCHOCKI, L. R. et al. Sicret - sistema de cruzamento de registros telefônicos. In: SOCIEDADE BRASILEIRA DE COMPUTAçãO. *XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2013*. Porto Alegre, 2013. p. 551. Instituto de Criminalística do Paraná, Pontifícia Universidade Católica do Paraná (PUCPR/PPGIa).

GROOTENDORST, M. *KeyBERT: Minimal keyword extraction with BERT.* Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.4461265>.

HASSAN, N. A. *Digital Forensics Basics: A Practical Guide Using Windows OS*. New York, New York, USA: Apress, 2019. ISBN 978-1-4842-3837-0. Disponível em: <https://doi.org/10.1007/978-1-4842-3838-7>.

HO, T. K. Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 1995. v. 1, p. 278–282 vol.1.

HOLTZMAN, A. et al. *The Curious Case of Neural Text Degeneration*. 2020. Disponível em: <https://arxiv.org/abs/1904.09751>.

HU, C. et al. *Unveiling the Potential of Knowledge-Prompted ChatGPT for Enhancing Drug Trafficking Detection on Social Media*. 2023. Disponível em: <https://arxiv.org/abs/2307.03699>.

HU, C. et al. Fine-grained classification of drug trafficking based on instagram hashtags. *Decision Support Systems*, v. 165, p. 113896, 2023. ISSN 0167-9236. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167923622001671>.

HU, C. et al. *Identifying Illicit Drug Dealers on Instagram with Large-scale Multimodal Data Fusion*. 2021. Disponível em: <https://arxiv.org/abs/2108.08301>.

HUANG, W. et al. An empirical study of llama3 quantization: from llms to mllms. *Visual Intelligence*, Springer Science and Business Media LLC, v. 2, n. 1, dez. 2024. ISSN 2731-9008. Disponível em: <http://dx.doi.org/10.1007/s44267-024-00070-x>.

JAMES, G. et al. *An Introduction to Statistical Learning: with Applications in R*. New York, NY: Springer, 2013. ISBN 978-1-4614-7137-0.

KILIÇ, R. *Linear and nonlinear econometric models against machine learning models: realized volatility prediction*. [S.l.], 2025. Disponível em: <https://www.federalreserve.gov/econres/feds/files/2025061pap.pdf>.

KOCHNEV, R. et al. *Optuna vs Code Llama: Are LLMs a New Paradigm for Hyperparameter Tuning?* 2025. ArXiv:2504.06006v1.

KUBAT, M. *An Introduction to Machine Learning*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2017.

LACERDA, C. R. B. *Classificação Supervisionada: Árvores de Decisão e Florestas Aleatórias*. Dissertação (Dissertação de Mestrado) — Universidade de Coimbra, Coimbra, Setembro 2023.

LACERDA, T. et al. Deep learning and mel-spectrograms for physica violence detection in audio. In: *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2021. p. 268–279. ISSN 2763-9061. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/18259>.

LAI, L.-H. et al. The use of machine learning models with optuna in disease prediction. *Electronics*, v. 13, n. 23, 2024. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/13/23/4775>.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics*, v. 33, n. 1, p. 159–174, March 1977.

LIANG, W.; LIANG, Y. *BPDec: Unveiling the Potential of Masked Language Modeling Decoder in BERT pretraining*. 2024. Disponível em: <https://arxiv.org/abs/2401.15861>.

LIU, X.; WANG, C. An empirical study on hyperparameter optimization for fine-tuning pre-trained language models. In: ZONG, C. et al. (Ed.). *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021. p. 2286–2300. Disponível em: <https://aclanthology.org/2021.acl-long.178/>.

LUZ, F. F. *Deep neural semantic parsing: translating from natural language into SPARQL*. Tese (Tese (Doutorado em Ciência da Computação)) — Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019. Acesso em: 2023-11-16.

MARTINS, V. et al. Comparing pocketsphinx and vosk recognition in human speech decoding. In: *Proceedings of IV Brazilian Humanoid Robot Workshop (BRAHUR) and V Brazilian Workshop on Service Robotics (BRASERO)*. Salvador(BA): UNEB, 2021. Accessed: 2024-06-13. Disponível em: <https://www.even3.com.br/anais/brahurbrasero/384168-COMPARING-POCKETSPHINX-AND-VOSK-RECOGNITION-IN-HUMAN-SPEECH-DECODING>.

MCQUADE, S. C. *Encyclopedia of Cybercrime*. Westport, CT: Greenwood Press, 2009. (Library of Congress Cataloging-in-Publication Data). ISBN 978-0-313-33975-4. Disponível em: <http://www.greenwood.com>.

MEIRELLES, F. *Cidade de Deus*. Rio de Janeiro, RJ: O2 Filmes e Vídeo Filmes, 2002. Film.

MENDES, A. Linguística de corpus e outros usos do corpus em linguística. In: MARTINS, A. M.; CARRILHO, E. (Ed.). *Manual de Linguística Portuguesa*. Berlin/Boston: Walter de Gruyter, 2016. p. 224–251.

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into texts. *Stroudsburg, Pennsylvania*, Jul 2004. Disponível em: <https://digital.library.unt.edu/ark:/67531/metadc30962/>.

MINAEE, S. et al. Deep learning based text classification: A comprehensive review. *CoRR*, abs/2004.03705, 2020. Disponível em: <https://arxiv.org/abs/2004.03705>.

MITCHELL, T. M. *Machine Learning*. New York, NY, USA: McGraw-Hill Science/Engineering/Math, 1997. 432 p. ISBN 0070428077.

MOONS, F.; VANDERVIEREN, E. *Measuring agreement among several raters classifying subjects into one-or-more (hierarchical) nominal categories. A generalisation of Fleiss' kappa*. 2023. Disponível em: <https://arxiv.org/abs/2303.12502>.

MOURãO, V. D. G. *Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)*. Dissertação (Dissertação de Mestrado) — Universidade de Brasília, Brasília, 2022. Orientador: Prof. Dr. Daniel Oliveira Cajueiro. Disponível em: <http://repositorio2.unb.br/jspui/handle/10482/44831>.

MULAKALA, B. et al. Adaptive multi-fidelity hyperparameter optimization in large language models. In: *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*. [S.l.: s.n.], 2024. p. 1–5.

NAKAISHI, K. et al. *Critical Phase Transition in Large Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2406.05335>.

NASCIMENTO, R. F. F. et al. O algoritmo support vector machines (svm): avaliação da separação ótima de classes em imagens ccd-cbers-2. In: INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. *Anais XIV Simpósio Brasileiro de Sensoriamento Remoto*. Natal, Brasil: INPE, 2009. p. 2079–2086.

NAVES, M. M. L. Estudo dos fatores interferentes no processo de análise de assunto. *Perspectivas em Ciência da Informação*, Belo Horizonte, v. 6, n. 2, p. 189–203, jul./dez. 2001.

OGURI, P. *APRENDIZADO DE MÁQUINA PARA O PROBLEMA DE SENTIMENT CLASSIFICATION*. Dissertação (Dissertação (mestrado)) — Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, 2006. Orientador: Ruy Luiz Milidiú; Co-orientador: Raúl Rentería. Disponível em: <https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=9947@1>.

PARANÁ, P. C. do. Polícia científica do paraná lança exposição itinerante sobre perícia no mundo digital. *Polícia Científica do Paraná*, p. 1–7, 2023. Disponível em: <https://www.pc.pr.gov.br/portal/noticias/policia-cientifica-do-parana-lanca-exposicao-itinerante-sobre-pericia-no-mundo-digital>.

POLÍCIA CIENTÍFICA DO ESTADO DO PARANÁ. *Polícia Científica e Polícia Civil propõem compra de equipamentos específicos*. 2023. <https://www.policiacientifica.pr.gov.br/Noticia/Policia-Cientifica-e-Policia-Civil-propoem-compra-de-equipamentos-especificos-para>. Acesso em: 23/08/2023 – 21:17h.

POLÍCIA CIENTÍFICA DO ESTADO DO PARANÁ. *Produtividade e Desempenho*. 2023. <https://www.policiacientifica.pr.gov.br/Pagina/Produtividade-e-Desempenho>. Computação Forense > Exames solicitados (peças), Acesso em: 23/08/2023 – 22:05h.

POVEY, D. et al. The kaldi speech recognition toolkit. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. [S.l.]: IEEE Signal Processing Society, 2011. IEEE Catalog No.: CFP11SRW-USB.

Priberam. *Dicionário Priberam da Língua Portuguesa - Entrada para "Frase"*. Priberam, 2024. <https://dicionario.priberam.org/frase>. Acessado em 04 de fevereiro de 2024. Disponível em: <https://dicionario.priberam.org/frase>.

Priberam. *Dicionário Priberam da Língua Portuguesa - Entrada para "Texto"*. Priberam, 2024. <https://dicionario.priberam.org/texto>. Acessado em 04 de fevereiro de 2024. Disponível em: <https://dicionario.priberam.org/texto>.

RADFORD, A. et al. Robust speech recognition via large-scale weak supervision. In: *Proceedings of the 40th International Conference on Machine Learning*. [S.l.]: JMLR.org, 2023. (ICML'23).

RAULINO, T. et al. Classification and association rules in brazilian supreme court judgments on pre-trial detention author proof. In: . [S.l.: s.n.], 2021. ISBN 978-3-030-86610-5.

RIBEIRO, L. S. F. *Cross Domain Visual Search with Feature Learning using Multi-stream Transformer-based Architectures*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023. Acesso em: 2023-10-08.

ROCHA, D. S. C. *Aprendizado de Máquina Aplicado ao Reconhecimento Automático de Falhas em Máquinas Rotativas*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Minas Gerais, Escola de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica, Belo Horizonte, MG, Brasil, June 2018. Orientador: Prof. João Antônio de Vasconcelos.

RODRIGUES, F. B. et al. Natural language processing applied to forensics information extraction with transformers and graph visualization. *IEEE Transactions on Computational Social Systems*, p. 1–17, 2022.

ROSE, S. et al. Automatic keyword extraction from individual documents. In: ____. [S.l.: s.n.], 2010. p. 1 – 20. ISBN 9780470689646.

ROSENFELD, R. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, v. 88, n. 8, p. 1270–1278, 2000.

SAMMUT, C.; WEBB, G. (Ed.). *Encyclopedia of Machine Learning*. Berlin: Springer, 2010.

SANTOS, W. et al. Comparing prompt-based llms, fine-tuning, and classical models for legal text classification in portuguese. In: *Anais do XXII Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2025. p. 1138–1149. ISSN 2763-9061. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/38798>.

SCALERCIO, A. M. R. D. A. et al. Evaluating LLMs for Portuguese sentence simplification with linguistic insights. In: CHE, W. et al. (Ed.). *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, 2025. p. 24452–24477. ISBN 979-8-89176-251-0. Disponível em: <https://aclanthology.org/2025.acl-long.1193/>.

SHAH, N. et al. An unsupervised machine learning approach for the detection and characterization of illicit drug-dealing comments and interactions on instagram. *Substance Abuse*, Taylor & Francis, v. 43, n. 1, p. 273–277, 2022. Epub 2021 Jul 2.

SILVA, M. et al. Evaluating domain-adapted language models for governmental text classification tasks in portuguese. In: *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*. Porto Alegre, RS, Brasil: SBC, 2024. p. 247–259. ISSN 2763-8979. Disponível em: <https://sol.sbc.org.br/index.php/sbbd/article/view/30697>.

SOUZA, F. et al. Bertimbau: Pretrained bert models for brazilian portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). *Intelligent Systems*. Cham: Springer International Publishing, 2020. p. 403–417. ISBN 978-3-030-61377-8.

SOUZA, F. C. d. *BERTimbau: pretrained BERT models for brazilian portuguese = BERTimbau: modelos BERT pré-treinados para português brasileiro*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2020. 1 recurso online. Disponível em: <https://hdl.handle.net/20.500.12733/1640809>.

SUN, S. et al. Capturing global informativeness in open domain keyphrase extraction. In: WANG, L. et al. (Ed.). *Natural Language Processing and Chinese Computing*. Cham: Springer International Publishing, 2021. p. 275–287. ISBN 978-3-030-88483-3.

SUSANDRI, S. et al. Enhancing text sentiment classification with hybrid cnn-bilstm model on whatsapp group. *Journal of Advances in Information Technology (JAIT)*, Engineering and Technology Publishing, v. 15, n. 3, p. 355–363, 2024. Published March 14, 2024.

TEJ, N. N. *Mastering Perplexity: A Comprehensive Guide to Understanding and Using Perplexity in AI and NLP*. AI Technology Suite, 2024. 32 p. Ranking: #151,591 em Loja Kindle, #1,288 em Computação, internet e mídia digital em inglês. ISBN B0DKWX954Z. Disponível em: <https://www.amazon.com/dp/B0DKR7XVRJ>.

TRIBES, C. et al. *Hyperparameter Optimization for Large Language Model Instruction-Tuning*. 2024. Disponível em: <https://arxiv.org/abs/2312.00949>.

VAPNIK, V. N. *The nature of statistical learning theory*. 2nd ed.. ed. New York: Springer, 1999. cm p. (Statistics for engineering and information science). Includes bibliographical references and index. Printed on acid-free paper. ISBN 0-387-98780-0.

VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.

VIEIRA, S. L.; CRUZ, A. R. d. Perícia forense computacional em telefones celulares com sistema operacional android. *Segurança, Justica e Cidadania*, v. 10, n. 2, p. 69–86, 2016. Disponível em: <https://www.gov.br/mj/pt-br/assuntos/sua-seguranca/seguranca-publica/analise-e-pesquisa/download/estudos/sjcvolume9/pericia_forense_computacional_telefones_celulares_sistema_operacional_android.pdf>.

WIJNBERG, D.; LE-KHAC, N.-A. Identifying interception possibilities for whatsapp communication. *Forensic Science International: Digital Investigation*, v. 38, p. 301132, 2021. ISSN 2666-2817. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666281721000305>.

WONGVORACHAN, T. et al. A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining. *Information*, v. 14, p. 54, 01 2023.

YANG, X.; LUO, J. *Tracking Illicit Drug Dealing and Abuse on Instagram using Multimodal Analysis*. 2016. Disponível em: <https://arxiv.org/abs/1605.02710>.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

ZHAO, W. X. et al. *A Survey of Large Language Models*. 2023. Disponível em: <http://arxiv.org/abs/2303.18223>.