

UNIVERSIDADE FEDERAL FLUMINENSE

ARTHUR BITTENCOURT VASCONCELOS

Balanceamento de Classes com Seleção de Paráfrases
para Classificação Binária de Textos

NITERÓI

2026

ARTHUR BITTENCOURT VASCONCELOS

Balanceamento de Classes com Seleção de Paráfrases
para Classificação Binária de Textos

Dissertação apresentada ao Programa
de Pós-Graduação em Computação da
Universidade Federal Fluminense como
requisito parcial para a obtenção do Grau
de Mestre em Computação. Área de
concentração: Ciência da Computação

Orientadora:

ALINE MARINS PAES CARVALHO

Coorientadores:

**ALEXANDRE PLASTINO DE CARVALHO
WASHINGTON LUIZ MIRANDA DA CUNHA**

NITERÓI

2026

Ficha catalográfica automática - SDC/BCPV
Gerada com informações fornecidas pelo autor

V331b Vasconcelos, Arthur Bittencourt
Balanceamento de Classes com Seleção de Paráfrases para
Classificação Binária de Textos / Arthur Bittencourt
Vasconcelos. - 2026.
77 f.: il.

Orientador: Aline Marins Paes Carvalho.
Coorientador: Alexandre Plastino De Carvalho; Washington
Luiz Miranda Da Cunha.
Dissertação (mestrado)-Universidade Federal Fluminense,
Instituto de Computação, Niterói, 2026.

1. Processamento de linguagem natural (Computação). 2.
Inteligência artificial. 3. Aprendizado de máquina. 4.
Produção intelectual. I. Marins Paes Carvalho, Aline,
orientadora. II. Plastino De Carvalho, Alexandre,
coorientador. III. Luiz Miranda Da Cunha, Washington,
coorientador. IV. Universidade Federal Fluminense. Instituto
de Computação. V. Título.

CDD - XXX

ARTHUR BITTENCOURT VASCONCELOS

Balanceamento de Classes com Seleção de Paráfrases para Classificação
Binária de Textos

Dissertação apresentada ao Programa de
Pós-Graduação em Computação da Univer-
sidade Federal Fluminense como requisito
parcial para a obtenção do Grau de Mes-
tre em Computação. Área de concentração:
Ciência da Computação

Aprovada em Agosto de 2026.

BANCA EXAMINADORA

Profa. ALINE MARINS PAES CARVALHO - Orientadora, UFF

Prof. ALEXANDRE PLASTINO DE CARVALHO - Coorientador, UFF

Prof. WASHINGTON LUIZ MIRANDA DA CUNHA - Coorientador, UNICAMP

Profa. MARIZA FERRO, UFF

Prof. EDUARDO BEZERRA DA SILVA, CEFET/RJ

Niterói

2026

Agradecimentos

Aos meus orientadores, aprecio de coração todo o tempo, dedicação e pelo aprendizado que obtive nessa pesquisa e durante o mestrado. Obrigado por estarem junto comigo nos erros, e por me ensinar a chegar nos acertos. Aos meus pais por possibilitarem o privilégio de focar apenas nos estudos. E aos meus amigos que apoiaram durante o mestrado, sendo um ombro, opinando nas minhas teorias, e sendo boas companhias.

Resumo

O desbalanceamento de classes é um desafio recorrente na tarefa de classificação de textos. Embora estratégias tradicionais sejam aplicadas para mitigar o problema, a área tem evoluído para métodos baseados em Modelo de Linguagem de Larga Escala, do inglês *Large Language Model* (LLM). Todavia, a falta de comparações amplas entre as diferentes abordagens levanta dúvidas quanto à sua vantagem real em relação aos métodos tradicionais. Este trabalho investiga os métodos de balanceamento em duas etapas. Na primeira, realiza um estudo que compara abordagens tradicionais de balanceamento, como SMOTE e EDA em relação a abordagens baseadas em LLMs, como PREDATOR e paráfrases geradas por LLMs. Avaliando o efeito da aplicação dessas técnicas em classificadores lineares e em modelos modernos baseados na arquitetura *Transformers*. Os resultados dessa etapa demonstram que a geração indiscriminada de paráfrases não se traduz em ganho preditivo consistente, alcançando o melhor desempenho apenas em uma das nove bases avaliadas. Por outro lado, o método representativo da abordagem baseada em LLMs, o PREDATOR, apresenta vantagem marginal em relação aos demais métodos, o que indica que a inclusão de um filtro de qualidade pode ser o diferencial do método. Portanto, a presente dissertação propõe um método de aumento de dados denominado Aumento por Paráfrases Seleccionadas (APS). O método utiliza a capacidade generativa de LLMs para criar sentenças e introduz mecanismos heurísticos de seleção baseados em duas vertentes: a primeira vertente fundamenta-se na similaridade semântica entre o exemplo gerado e os exemplos minoritários da base, priorizando as paráfrases semanticamente mais próximas da representação central da classe minoritária real. A segunda vertente baseia-se na amostragem com base na incerteza oriunda do aprendizado ativo, que sugere que os exemplos mais valiosos para o treinamento do modelo são justamente aqueles que o classificador apresenta maior dificuldade em categorizar. A avaliação do APS demonstra que a inserção do mecanismo de seleção incrementa a efetividade do método proposto, alcançando melhorias nos valores absolutos de desempenho preditivo (em média 11,28% de macro-F1) e atingindo o topo no *ranking* de desempenho entre os métodos avaliados.

Palavras-chave: Classificação de Textos. Desbalanceamento de Classes. Modelos de Linguagem de Grande Porte. Aumento de Dados.

Abstract

Class imbalance is a recurring challenge in the text classification task. Although traditional strategies such as SMOTE and EDA are applied to mitigate this problem, the field has been evolving toward methods based on Large Language Models (LLMs). However, the lack of comprehensive comparisons among approaches casts doubt on their real advantage over traditional methods. This work investigates class balancing methods in two stages. In the first stage, it presents a study comparing traditional balancing approaches, such as SMOTE and EDA, with LLM-based approaches, such as PREDATOR and LLM-generated paraphrases, and evaluates the impact of applying these techniques to linear classifiers and modern *Transformer*-based models. The results of this stage demonstrate that the indiscriminate generation of paraphrases does not yield consistent predictive gains, with top predictive performance achieved in only one of the nine evaluated datasets. On the other hand, PREDATOR shows a marginal advantage over other methods, suggesting that the inclusion of a quality filter may be the key differentiator. Therefore, this dissertation proposes a data augmentation method called Augmentation by Paraphrase Selection (APS, or *Aumento por Paráfrases Seleccionadas* in Portuguese). The method leverages the generative capability of LLMs to synthesize sentences and introduces heuristic selection mechanisms based on two fronts: the first front is grounded in the semantic similarity between the generated sample and the minority instances in the dataset, prioritizing paraphrases that are semantically closest to the central representation of the actual minority class. The second front relies on uncertainty sampling derived from active learning, which posits that the most valuable samples for model training are precisely those that the classifier finds most difficult to categorize. The evaluation of APS demonstrates that the incorporation of the selection mechanism alters the effectiveness of the method, achieving improvements in absolute predictive performance metrics (11,28% on average macro-F1) and ranking among the top methods evaluated.

Keywords: Text Classification. Class Imbalance. Large Language Models. Data Augmentation.

Lista de Figuras

1	Resultado do processo de desbalanceamento artificial. A razão escolhida para cada base foi destacada por uma estrela.	32
2	Arquitetura geral do PREDATOR.	35
3	Arquitetura geral do método de paráfrases geradas por LLMs.	36
4	Arquitetura geral do método proposto APS.	45
5	Método de seleção por similaridade ao vetor protótipo.	49
6	Método de seleção por entropia.	51
7	Análise <i>post-hoc</i> de Nemenyi entre métodos de seleção de paráfrase (LR). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.	57
8	Análise <i>post-hoc</i> de Nemenyi entre métodos de balanceamento (LR). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.	58
9	Análise <i>post-hoc</i> de Nemenyi entre métodos de seleção de paráfrase (XLM-R). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.	60
10	<i>Trade-off</i> desempenho preditivo vs. tempo em log segundos (LR).	65
11	<i>Trade-off</i> desempenho preditivo vs. tempo em log segundos (XLM-R). . .	65

Lista de Tabelas

1	Comparativo de trabalhos relacionados em sobreamostragem e TDA. . . .	29
2	Resumo das estatísticas das bases de dados avaliadas.	31
3	Resultados da Regressão Logística para as nove bases.	39
4	Resultados do <i>fine-tuning</i> do XLM-R para as nove bases.	40
5	Estratégia com melhor desempenho médio por base.	41
6	Exemplos gerados pelas técnicas vencedoras em cada base.	42
7	Desempenho preditivo entre seletores de paráfrase com $\alpha = 0,5$ (LR) . . .	55
8	Desempenho preditivo entre seletores de paráfrase com $\alpha = 1,0$ (LR) . . .	56
9	Desempenho preditivo entre métodos de TDA com $\alpha = 0,5$ (LR)	56
10	Desempenho preditivo entre métodos de TDA com $\alpha = 1,0$ (LR)	56
11	Análise entre seletores de paráfrase com $\alpha = 0,5$ (XLM-R)	59
12	Análise entre seletores de paráfrase com $\alpha = 1,0$ (XLM-R)	59
13	Desempenho preditivo entre métodos de TDA com $\alpha = 0,5$ (XLM-R) . . .	59
14	Desempenho preditivo entre métodos de TDA com $\alpha = 1,0$ (XLM-R) . . .	60
15	Exemplos gerados pelas técnicas vencedoras do estudo anterior em comparação com a técnica vencedora Paráfrase selecionada por APS.	61
16	Exemplos gerados pelas técnicas vencedoras nas demais bases.	63
17	Tempo total para a geração de exemplos em segundos.	64
18	Resumo dos melhores resultados por base de dados.	66

Lista de Abreviaturas e Siglas

APS Aumento por Paráfrases Seleccionadas

CT Classificação de Textos

EDA *Easy Data Augmentation*

LLM Modelo de Linguagem de Larga Escala, do inglês *Large Language Model*

PLN Processamento de Linguagem Natural

SMOTE *Synthetic Minrurity Oversampling TEchnique*

TDA Aumento de Dados Textual, do inglês *Text Data Augmentation*

Sumário

1	Introdução	13
2	Fundamentação Teórica	18
2.1	Classificação de Textos	18
2.2	Desbalanceamento de Classes	20
2.3	Métricas de Classificação	22
3	Trabalhos Relacionados	24
3.1	Métodos de Aumento de Dados Textual	24
3.2	Métodos de Sobreamostragem Textual	26
3.3	Conclusões	28
4	Avaliação de Técnicas de Sobreamostragem Textual	30
4.1	Bases de Dados	30
4.1.1	Bases Artificialmente Desbalanceadas	30
4.1.2	Bases com Desbalanceamento Natural ou Decorrente da Binarização	33
4.2	Métodos de Aumento de Dados Avaliados	33
4.2.1	SMOTE	34
4.2.2	EDA	34
4.2.3	PREDATOR	35
4.2.4	Paráfrases Geradas por LLMs	36
4.3	Estratégias de Classificação	38
4.4	Resultados	39

4.5	Discussão	43
5	Paráfrases com LLMs e Seletores	
	Heurísticos	44
5.1	Arquitetura Geral do Método	44
5.2	Paráfrase por meio de LLMs	46
5.3	Seleção de Paráfrases	47
5.3.1	Vetor Protótipo	48
5.3.2	Entropia de Shannon	50
6	Avaliação do Método de Aumento por Paráfrases Seleccionadas	53
6.1	Metodologia Experimental	53
6.1.1	<i>Baselines</i> de Seleção de Paráfrase	53
6.1.2	Análise Quantitativa de Exemplos Gerados	54
6.2	Análise de Desempenho Preditivo	55
6.2.1	Desempenho com LR	55
6.2.2	Desempenho com XLM-R	58
6.3	Análise dos Exemplos Gerados	61
6.4	Análise de Custo	63
6.5	Sumarização dos Resultados	66
7	Conclusões e Trabalhos Futuros	68
7.1	Conclusões	68
7.2	Limitações	69
7.3	Trabalhos Futuros	70
7.4	Considerações Éticas	71
	REFERÊNCIAS	72

1 Introdução

Classificação de Textos (CT) é uma das tarefas mais tradicionais do Processamento de Linguagem Natural (PLN) e frequentemente é formulada como um problema de aprendizado supervisionado, no qual modelos são treinados para associar categorias previamente definidas a sequências textuais (Minaee *et al.*, 2021). Contudo, o desbalanceamento no número de instâncias por classe semântica permanece um desafio recorrente em tarefas de CT, especialmente em cenários reais, em que a distribuição das categorias tende a ser naturalmente desigual (Gao *et al.*, 2026; Bayer; Kaufhold; Reuter, 2022).

Em diversas aplicações de CT, como análise de sentimentos, detecção de discurso de ódio, filtragem de *spam* e classificação de documentos médicos (Jang *et al.*, 2021), é comum que a classe de interesse esteja sub-representada no conjunto de dados, caracterizando um cenário de desbalanceamento de classe (He; Garcia, 2009; Bayer; Kaufhold; Reuter, 2022). Neste cenário, modelos de aprendizado de máquina tendem a favorecer a classe majoritária durante o treinamento, uma vez que ela domina a distribuição dos dados. Como consequência, o modelo torna-se enviesado em favor dessa classe, apresentando elevado desempenho na classe majoritária, mas desempenho inferior na classe minoritária (Gao *et al.*, 2026).

Dentre as estratégias para mitigar o desbalanceamento de classes, destaca-se o rebalanceamento de dados, seja por meio da geração de instâncias da categoria sub-representada (sobreamostragem) (Gao *et al.*, 2026), ou pela remoção de exemplos da classe majoritária (subamostragem) (Fonseca *et al.*, 2025). No escopo da sobreamostragem de instâncias em CT, a literatura distingue duas abordagens fundamentais: as técnicas de amostragem clássicas e os métodos baseados em Aumento de Dados Textual, do inglês *Text Data Augmentation* (TDA) (Bayer; Kaufhold; Reuter, 2022).

As técnicas clássicas, como o *Synthetic Minority Oversampling TEchnique* (SMOTE) (Chawla *et al.*, 2002), operam no espaço de características, interpolando representações numéricas após a vetorização do texto. Por outro lado, as técnicas de TDA, como o *Easy*

Data Augmentation (EDA) (Wei; Zou, 2019), atuam diretamente na superfície do texto. Em geral, essas abordagens expandem o conjunto de exemplos da classe minoritária por meio de perturbações heurísticas ou substituições semânticas, enriquecendo a variabilidade lexical e contextual dos exemplos de treinamento.

Contudo, por operarem de maneira isolada sobre palavras individuais, técnicas tradicionais de TDA baseadas em heurísticas locais correm o risco de corromper a semântica global do documento ou induzir o fenômeno de inversão de rótulo (*label flipping* (Natarajan *et al.*, 2013)), gerando instâncias ruidosas que podem, potencialmente, degradar o aprendizado do classificador (Anaby-Tavor *et al.*, 2020). Em resposta a essas limitações estruturais, o uso de Modelo de Linguagem de Larga Escala, do inglês *Large Language Model* (LLM) tem sido apresentado como uma direção promissora na literatura de sobreamostragem textual. Como evidenciado em (Patil *et al.*, 2022; Radliński; Guściora; Kocoń, 2025), LLMs demonstram habilidades adequadas para gerar exemplos coerentes com o contexto e compatíveis com o rótulo original, superando a rigidez das abordagens baseadas em perturbações superficiais.

São exemplos desse novo paradigma a geração de exemplos e filtragem, adotada pelo PREDATOR (Abonizio; Junior, 2020) e a geração de paráfrases por meio de LLMs (Radliński; Guściora; Kocoń, 2025). O primeiro método combina um modelo gerador e um modelo classificador discriminativo, ambos ajustados à base. De forma complementar, a geração de paráfrases por meio de LLMs se vale da capacidade destes modelos de criar exemplos preservando a significado essencial dos exemplos presentes na base de treinamento, sem precisar ajustar o modelo.

Por outro lado, a literatura recente valida essas técnicas de TDA baseadas em LLMs em cenários limitados, restringindo-se a poucas bases de dados (uma base em (Radliński; Guściora; Kocoń, 2025; Patil *et al.*, 2022) e três bases em (Abonizio; Paraiso; Barbon, 2022)) ou a domínios específicos, como análise de sentimentos (Abonizio; Paraiso; Barbon, 2022). Em termos práticos, a aplicação de LLMs para criação de exemplos sintéticos levanta dúvidas sobre a real utilidade e/ou representatividade dessas instâncias, uma vez que o esforço computacional despendido nem sempre se traduz em um aprendizado efetivo para o classificador (Min *et al.*, 2023).

Diante desse cenário, faz-se necessária uma investigação que valide estes métodos de sobreamostragem textual de forma mais ampla, comparando seus efeitos em cenários variados de desbalanceamento e abrangendo diversos domínios textuais. Define-se, então, a primeira questão de pesquisa desta dissertação (**QP1**): *Métodos de sobreamostragem*

textual baseados em LLM são mais eficazes para mitigar o problema do desbalanceamento de classes em comparação com abordagens tradicionais?

Para responder a essa pergunta de pesquisa, formalizando a primeira contribuição desta dissertação, realiza-se uma avaliação do ganho de desempenho de métodos clássicos de sobreamostragem (SMOTE) e métodos tradicionais de TDA (EDA) em relação aos métodos de TDA baseados em LLMs (PREDATOR e paráfrases geradas por LLMs) em nove bases de diferentes domínios. Para viabilizar esse comparativo, uma outra contribuição desta dissertação é a introdução de bases artificialmente desbalanceadas, criando cenários controlados onde o desbalanceamento de classes é o principal desafio da classificação.

Os resultados dessa investigação demonstram que dentre ambos os métodos representativos da abordagem de TDA baseada em LLMs, o PREDATOR gera exemplos frequentemente truncados, enquanto o método de paráfrases geradas por LLMs resulta em textos coerentes. Porém, esse último método apresenta limitações metodológicas claras. A inserção de novos exemplos sem um critério de triagem explícito atinge um ganho de 19% de desempenho em relação à base desbalanceada em apenas uma das nove bases testadas. O outro método representativo do paradigma de TDA, PREDATOR, atingiu a melhor e a segunda melhor posição quando avaliado pelo *ranking* médio de desempenho relativo nos dois cenários investigados. Embora a sua vantagem em relação aos demais métodos seja pequena, os resultados do PREDATOR indicam que a inclusão de um filtro de qualidade pode ser o diferencial do método, oferecendo um desempenho marginalmente superior em relação aos demais métodos de balanceamento. A partir desta hipótese, define-se a segunda questão de pesquisa desta dissertação (**QP2**): *Qual é o impacto de diferentes estratégias de filtragem impacta o desempenho do aumento de dados baseado em paráfrases na mitigação do desbalanceamento de classes?*

Para responder a essa segunda questão de pesquisa e com o objetivo de superar as limitações identificadas na aplicação de paráfrases para o aumento de dados em cenários desbalanceados, mantendo os benefícios da abordagem, esta dissertação propõe o método Aumento por Paráfrases Seleccionadas (APS) como principal contribuição. O método se vale da capacidade de generalização de LLMs para criar paráfrases sem fazer o ajuste fino no domínio dos dados e incorpora uma camada heurística de seleção para filtrar as instâncias mais promissoras. Embora o PREDATOR também combine um componente gerador com um filtro de qualidade, o método proposto difere-se principalmente no que diz respeito à estratégia de seleção de exemplos. Enquanto o PREDATOR filtra seus

exemplos com base na probabilidade de o exemplo gerado pertencer à classe de interesse, o método APS emprega seletores heurísticos para priorizar regiões no espaço de dados. Dessa forma, o método APS foi avaliado por meio de duas instâncias principais, que se distinguem pela vertente em que o mecanismo de seleção opera.

A primeira vertente fundamenta-se na similaridade semântica do exemplo gerado em relação aos exemplos minoritários da base, priorizando as paráfrases semanticamente mais próximas da representação central da classe minoritária real. A segunda vertente baseia-se na abordagem de amostragem por incerteza do aprendizado ativo (Settles, 2009), que se vale da premissa de que os exemplos mais valiosos para o treinamento do modelo são justamente aqueles em que o modelo discriminativo apresenta menor confiança na classificação, priorizando as paráfrases mais próximas da fronteira de decisão deste modelo.

A avaliação do método demonstra que ele apresenta desempenho preditivo superior em cinco das nove bases testadas. No entanto, a aplicação do método gera um aumento drástico no custo computacional em relação aos demais métodos de balanceamento, evidenciando, assim, um *trade-off* entre o ganho de desempenho preditivo e tempo de processamento. Portanto, os resultados obtidos reforçam o potencial da abordagem, ao mesmo tempo em que apontam oportunidades para pesquisas futuras voltadas à mitigação do custo computacional e à exploração de estratégias que tornem esse compromisso entre desempenho e eficiência mais favorável.

Os capítulos seguintes desta dissertação estão organizados da seguinte forma. O Capítulo 2 aborda a fundamentação teórica necessária à compreensão desta pesquisa, discutindo conceitos-chave, como a classificação de textos, o desbalanceamento e as métricas de desempenho de classificação. O Capítulo 3 apresenta os trabalhos relacionados, contextualizando as principais abordagens de aumento de dados e o uso de modelos de linguagem na literatura atual de sobreamostragem textual.

O Capítulo 4 detalha um estudo preliminar conduzido com o uso de LLMs para paráfrase. Neste capítulo, apresentam-se a metodologia experimental, a análise e a discussão dos resultados, que evidenciam as limitações da geração de dados sem critérios de filtragem, servindo de fundamentação empírica e de motivação para o método proposto. O Capítulo 5 introduz formalmente o método APS, descrevendo o processo de geração de paráfrase por meio de LLMs, os mecanismos de seleção desenvolvidos para mitigar os problemas identificados no estudo preliminar e a sumarização do método.

O Capítulo 6 apresenta a metodologia experimental e discute os resultados obtidos, incluindo a análise de desempenho preditivo, a avaliação qualitativa dos exemplos gerados

e a relação entre o tempo computacional despendido e o ganho de desempenho. Por fim, o Capítulo 7 apresenta as conclusões deste trabalho, destaca as principais contribuições e propõe caminhos para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo estabelece a base conceitual necessária para o desenvolvimento e a compreensão deste trabalho. A Seção 2.1 formaliza o problema de Classificação de Textos (CT) sob a ótica probabilística adotada nesta dissertação. A Seção 2.2 fundamenta o problema de desbalanceamento de classes e como ele afeta a tarefa de classificação. Por fim, a Seção 2.3 apresenta os critérios de avaliação e as métricas de desempenho empregados.

2.1 Classificação de Textos

De acordo com os autores em (Jurafsky; Martin, 2026), a tarefa de classificação pode ser formalizada como o aprendizado de uma função $f : \mathbb{R}^d \rightarrow Y$ que mapeia um vetor numérico de atributos $\mathbf{z} \in \mathbb{R}^d$ para uma categoria em um espaço discreto de classes $Y = \{y_1, \dots, y_n\}$, $n \geq 2$. Uma das abordagens para o problema de classificação baseia-se em modelos discriminativos, cujo objetivo é aproximar a probabilidade condicional $P(y_i | \mathbf{z})$, que representa a probabilidade de a entrada $\mathbf{z}$ pertencer à classe y_i . Durante a etapa de inferência, o modelo atribui à entrada o rótulo correspondente à classe com maior probabilidade condicional, seguindo a regra da máxima probabilidade a posteriori (MAP, do inglês *Maximum a Posteriori*). Formalmente, o rótulo predito $\hat{y}$ é obtido por meio da função *arg max*:

$$\hat{y} = \arg \max_{y_i \in Y} P(y_i | \mathbf{z}) \quad (2.1)$$

Para aprender essa distribuição condicional, os parâmetros do modelo são ajustados por meio da otimização de uma função objetivo sobre um conjunto de dados de treinamento.

No contexto de CT, o dado de entrada é uma sequência textual $x \in X$, em que X representa o espaço de todas as possíveis sequências textuais. Nesse cenário, a classificação textual pode ser formulada como o mapeamento de uma função $f_{ct} : X \rightarrow Y$.

Diferentemente de dados tabulares ou numéricos, sequências textuais possuem natureza discreta, simbólica e de dimensão variável. Por esse motivo, antes de serem processadas por algoritmos de aprendizado de máquina, elas precisam ser convertidas em representações vetoriais contínuas $\mathbf{z} \in \mathbb{R}^d$, preservando, tanto quanto possível, as informações linguísticas relevantes. Nesse caso, o rótulo predito $\hat{y} \in Y$ é obtido por:

$$\hat{y} = f(E(x)) = f_{ct}(x) \quad (2.2)$$

Em que $E : X \rightarrow \mathbb{R}^d$ representa a função que codifica a sequência textual em uma representação vetorial $\mathbf{z} = E(x)$, posteriormente utilizada pelo classificador f .

Esse processo de mapeamento, conhecido como extração de atributos (*feature extraction*), pode variar de técnicas simples como contagem de palavras (e.g., *Term frequency* (TF)) (Manning; Raghavan; Schütze, 2008) a técnicas mais sofisticadas como a extração dos pesos de um modelo de linguagem pré-treinado baseado em *Transformer* (Vaswani et al., 2017).

As abordagens que se apoiam na contagem de palavras ou de termos (n -gramas) (Jurafsky; Martin, 2026) representam o documento por meio da frequência de seus termos. Apesar da simplicidade e do baixo custo computacional, essa estratégia gera vetores de alta dimensão e extrema esparsidade, além de ignorar a ordem das palavras, a estrutura sintática e o contexto semântico (Jurafsky; Martin, 2026).

Nesse contexto, a área de PLN evoluiu para utilizar representações vetoriais contínuas densas, em que os termos são projetados em espaços latentes contínuos de dimensão reduzida aprendidos de maneira auto-supervisionada a partir de um grande volume de textos. Inicialmente, os vetores gerados eram estáticos (*Word Embeddings*) (Mikolov et al., 2013; Pennington; Socher; Manning, 2014), tendo como modelo subjacente redes neurais rasas. Essas representações capturam relações de similaridade semântica por meio da hipótese distribucional. Contudo, por serem representações estáticas, atribuem um único vetor fixo a cada palavra do vocabulário, falhando em tratar fenômenos de polissemia (onde uma mesma palavra assume significados distintos a depender do contexto) (Peters et al., 2018).

O estado da arte da área é consolidado por modelos baseados na arquitetura *Transformer* (Vaswani et al., 2017). Fundamentados no mecanismo de atenção, esses modelos geram representações contextuais dinâmicas, nas quais a vetorização de um *token* é computada em função de toda a sequência ao seu redor. Modelos pré-treinados do tipo *encoder*

incorporam o processo de codificação do texto à própria arquitetura, produzindo representações vetoriais contextualizadas que podem ser ajustadas conjuntamente com a camada de classificação durante o *fine-tuning*, permitindo um treinamento fim a fim, para a tarefa de classificação e superando as limitações dos métodos predecessores, como a esparsidade e a polissemia (Devlin *et al.*, 2019; Conneau *et al.*, 2020).

2.2 Desbalanceamento de Classes

Em cenários práticos de classificação de textos, a distribuição de probabilidade marginal dos rótulos $P(y)$ raramente é uniforme (Gao *et al.*, 2026). Quando o desbalanceamento entre as classes é substancial, o processo de otimização do classificador é afetado, uma vez que a função de custo passa a ser dominada pelos erros associados às categorias mais frequentes. Como consequência, o ajuste do classificador é direcionado majoritariamente para a redução do erro dessas classes dominantes, resultando em um modelo com alto desempenho preditivo na classe majoritária, porém com baixa capacidade de generalização para a classe minoritária (Gao *et al.*, 2026).

O tratamento do desbalanceamento multiclasse ($|Y| > 2$) é significativamente mais complexo do que em problemas binários, em razão das diferentes proporções entre as classes e das possíveis relações existentes entre elas (Gao *et al.*, 2026). Dessa forma, esta dissertação restringe sua investigação ao cenário binário ($|Y| = 2$), permitindo uma análise mais controlada dos efeitos do desbalanceamento e da eficácia dos métodos propostos para mitigá-lo. Dessa forma, define-se o conjunto de treinamento original como $D = D_{maj} \cup D_{min}$, em que D_{maj} e D_{min} representam os subconjuntos de exemplos das classes majoritária e minoritária ($D_{min} \cap D_{maj} = \emptyset$), respectivamente, de modo que $|D_{maj}| \gg |D_{min}|$. O grau de desbalanceamento da base de dados é formalmente quantificado pela Razão de Desbalanceamento (RD), definida por:

$$RD = \frac{|D_{maj}|}{|D_{min}|} \quad (2.3)$$

Valores de $RD \approx 1$ indicam um cenário ideal de equilíbrio entre as classes, enquanto valores substancialmente maiores que 1 caracterizam cenários de desbalanceamento severo, nos quais o viés de otimização discutido anteriormente torna-se crítico para o aprendizado do classificador.

Para mitigar o problema de desbalanceamento de classes, uma das abordagens consiste

na intervenção em nível de dados por meio de técnicas de reamostragem. Essa abordagem divide-se majoritariamente entre métodos de subamostragem, que buscam remover instâncias da classe super-representada (D_{maj}), e estratégias de sobreamostragem, cujo propósito é gerar exemplos pertencentes à categoria minoritária (D_{min}) (Gao *et al.*, 2026).

Ao aplicar uma estratégia de sobreamostragem sobre o conjunto de treinamento original D , as novas instâncias geradas a partir da classe minoritária D_{min} compõem o conjunto sintético D_{syn} . O conjunto de treinamento final e ampliado, destinado ao ajuste do modelo, é definido pela união:

$$D_{aug} = D \cup D_{syn} \quad (2.4)$$

O objetivo final dessa estratégia é mitigar o viés do modelo em direção à classe majoritária, assumindo que o subconjunto original D_{min} é representativo em relação à distribuição real da classe minoritária. Matematicamente, o cenário ideal de balanceamento ocorre quando: $|D_{syn}| = |D_{maj}| - |D_{min}|$. Todavia, nem sempre é necessário balancear o conjunto perfeitamente (Chawla *et al.*, 2002). Com isso em mente, define-se um parâmetro α para delimitar a razão de desbalanceamento desejada, transformando o cenário objetivo em $|D_{syn}| = \alpha \times |D_{maj}| - |D_{min}|$. A razão de desbalanceamento resultante é RD_{aug} :

$$RD_{aug} = \frac{|D_{maj}|}{|D_{min}| + |D_{syn}|} = \frac{|D_{maj}|}{\alpha \times |D_{maj}|} = \frac{1}{\alpha} \quad (2.5)$$

O método proposto nesta dissertação se baseia também em conceitos de subamostragem como um passo intermediário. O objetivo da subamostragem é remover instâncias pouco representativas de D_{maj} para equilibrar o conjunto de treino. No método APS, esse processo foi adaptado para selecionar os exemplos de maior importância em D_{syn} resultando no conjunto de instâncias selecionadas D_{sel} . O mecanismo de seleção (detalhado no Capítulo 5) seleciona um número de exemplos tal que:

$$|D_{sel}| = \alpha \times |D_{maj}| - |D_{min}| \quad (2.6)$$

Transformando o processo da Equação 2.4 para:

$$D_{aug} = D \cup D_{sel} \quad (2.7)$$

2.3 Métricas de Classificação

Para avaliar o desempenho de um classificador, o modelo é submetido a um conjunto de dados previamente reservado, denominado conjunto de teste (D_{test}). O resultado desse processo pode ser representado pelo conjunto de trios $O = \{(x, \hat{y}, y) \mid (x, y) \in D_{test}, \hat{y} = f(E(x))\}$, em que cada elemento é composto pela instância x , pelo rótulo predito $\hat{y} \in Y$ e pelo respectivo rótulo verdadeiro $y \in Y$.

A partir desse conjunto de saídas, as predições são mapeadas em uma matriz de confusão, sendo categorizadas em quatro quadrantes fundamentais de acordo com a classe de interesse $y_i \in Y$, definindo a classe complementar como $y_j \in Y$ com $i \neq j$:

- **Verdadeiro Positivo (VP):** Casos em que o modelo previu a classe de interesse ($\hat{y} = y_i$) e o rótulo real correspondia a ela ($y = y_i$).
- **Falso Positivo (FP):** Casos em que o modelo previu a classe de interesse ($\hat{y} = y_i$), mas o rótulo real pertencia à classe complementar ($y = y_j$).
- **Verdadeiro Negativo (VN):** Casos em que o modelo previu a classe complementar ($\hat{y} = y_j$) e o rótulo real correspondia a ela ($y = y_j$).
- **Falso Negativo (FN):** Casos em que o modelo previu a classe complementar ($\hat{y} = y_j$), mas o rótulo real era a classe de interesse ($y = y_i$).

Em cenários de dados desbalanceados, a escolha de uma métrica adequada para avaliar o desempenho preditivo é crítica. A Acurácia (A) é uma métrica popular por sua simplicidade de compreensão e cálculo, definida pela razão entre os exemplos preditos corretamente e o número total de exemplos no conjunto de testes (D_{test}):

$$A = \frac{|VP| + |VN|}{|D_{test}|} \quad (2.8)$$

No entanto, em um cenário de desbalanceamento, a predição correta dos exemplos de maior proporção pode mascarar a incapacidade do modelo em classificar corretamente a classe minoritária, levando a conclusões equivocadas sobre a eficácia do sistema (He; Garcia, 2009; Gao *et al.*, 2026). Por exemplo, considere um conjunto de dados em que 99% dos exemplos pertencem à classe A e apenas 1% à classe B. Um modelo que sempre prediz a classe A alcançará 99% de acurácia, embora seja incapaz de identificar qualquer exemplo da classe B, tornando-se pouco útil na prática.

Para mitigar essa limitação da acurácia, a literatura adota métricas baseadas na proporção de erros e acertos dentro de subconjuntos específicos da matriz de confusão. A Precisão (P) mede a exatidão das predições positivas, indicando a proporção de instâncias classificadas como a classe de interesse que de fato pertencem a essa categoria. Matematicamente, seu cálculo é definido por:

$$P = \frac{|VP|}{|VP| + |FP|} \quad (2.9)$$

Por outro lado, a Revocação (R) quantifica a sensibilidade do classificador, ou seja, sua capacidade de capturar a totalidade das instâncias reais da classe e minimizar a ocorrência de falsos negativos (FN). A métrica é expressa da seguinte forma:

$$R = \frac{|VP|}{|VP| + |FN|} \quad (2.10)$$

Como frequentemente existe um compromisso envolvido entre otimizar a precisão ou a revocação do modelo, utiliza-se o *F1-Score* como uma métrica única que sintetiza ambas as perspectivas. O *F1-Score* é calculado por meio da média harmônica entre a Precisão e a Revocação:

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (2.11)$$

Como o *F1-Score* tradicional é calculado tendo uma classe específica como referência, adota-se a variante *Macro-F1* para agregar os resultados obtidos em todo o conjunto de dados (Minaee *et al.*, 2021), considerando todas as classes. No escopo da classificação binária abordada neste trabalho, o *Macro-F1* é calculado a partir da média aritmética simples dos valores de *F1-Score* medidos independentemente para cada uma das duas classes.

Essa abordagem garante que o desempenho na classe minoritária tenha o mesmo peso estatístico que o da classe majoritária, impedindo que a taxa de acerto global mascare cenários em que o classificador falhe em identificar a categoria sub-representada.

3 Trabalhos Relacionados

Este capítulo tem como objetivo situar a presente dissertação no panorama da literatura de métodos de TDA e de sobreamostragem textual. Para fornecer uma visão estruturada da evolução da área, a revisão foi dividida em duas partes: métodos de TDA (Seção 3.1), e métodos de sobreamostragem textual (Seção 3.2). A primeira parte apresenta um panorama geral de métodos de TDA e de estudos seminais da área, enquanto a segunda foca em trabalhos que apresentam métodos de sobreamostragem aplicados à área de PLN. Por fim, apresenta-se uma síntese do mapeamento deste capítulo (Seção 3.3), apontando as lacunas encontradas que serão preenchidas pela presente dissertação.

3.1 Métodos de Aumento de Dados Textual

Esta seção apresenta uma perspectiva geral da área de TDA, cujo objetivo geral é aumentar o número de exemplos na base por meio de transformações aplicadas aos exemplos existentes (Bayer; Kaufhold; Reuter, 2022). EDA (*Easy Data Augmentation*) (Wei; Zou, 2019) propõe a criação de novos exemplos por meio de operações heurísticas locais, como substituições por sinônimos, inserções, trocas e deleções aleatórias de palavras. Avaliado em bases consolidadas (como SST-2 (Socher *et al.*, 2013)), o EDA demonstrou que transformações de baixo custo computacional podem gerar ganhos preditivos expressivos, especialmente em cenários de escassez de recursos. No entanto, o estudo original não abordou configurações de desbalanceamento de classes e careceu de uma comparação direta de ganho preditivo contra métodos clássicos de reamostragem. Estudos que surgiram depois, avaliaram este método de TDA em cenários de desbalanceamento (Abonizio; Paraíso; Barbon, 2022) e serão abordados futuramente no presente capítulo.

Buscando superar a rigidez das heurísticas tradicionais presentes em (Wei; Zou, 2019), a literatura avançou rumo ao condicionamento contextual em nível de token. O principal marco dessa transição foi o método CBERT (*Conditional BERT*) (Wu *et al.*, 2019), que introduziu a perturbação textual baseada em modelos de linguagem mascarados. O

método realiza o ajuste fino do BERT (Devlin *et al.*, 2019), modificando sua arquitetura para receber o rótulo da classe como informação condicional, permitindo que o modelo substitua *tokens* mascarados por sinônimos contextualmente coerentes e semanticamente alinhados à categoria-alvo. Em avaliações conduzidas em múltiplas bases de dados, o CBERT demonstrou superioridade preditiva em relação a abordagens de substituição de sinônimos por meio da WordNet e a geradores condicionais baseados em arquiteturas recorrentes. Contudo, a validação do método concentrou-se estritamente em cenários de escassez de exemplos (*low-resource*), deixando lacunas quanto à sua eficácia prática em cenários de desbalanceamento.

Avançando para abordagens que operam transformações em nível de sentença, as técnicas de reescrita global ganharam espaço por meio de métodos de retrotradução (*back-translation*). Um estudo fundamental para a compreensão dessa dinâmica foi realizado por Edunov *et al.* (Edunov *et al.*, 2018), que formalizaram o processo de traduzir uma sentença para um idioma intermediário e, em seguida, revertê-la à sua língua original por meio de redes neurais baseadas em *Transformers* para gerar uma instância lexicalmente distinta, mas semanticamente equivalente. Os autores investigaram extensivamente diferentes estratégias de decodificação, demonstrando que a introdução de ruído amostral, combinada à busca por feixe (*beam search*), melhora a qualidade do texto gerado. Embora o estudo original tenha focado na modelagem de sistemas de tradução, a retrotradução consolidou-se como um método robusto de sobreamostragem, cuja eficácia na classificação textual em cenários de desbalanceamento de classes foi validada em estudos subsequentes, conforme mapeado em (Abonizio; Paraiso; Barbon, 2022).

A consolidação de arquiteturas de redes neurais profundas e de *Transformers*, como os modelos de tradução empregados na retrotradução, abriu caminho para o uso de modelos ainda mais robustos e de larga escala. O sucesso da reescrita global demonstrou que o uso de modelos generativos permitia transcender a substituição local de palavras, impulsionando a literatura em direção a abordagens inteiramente generativas baseadas em LLMs (Anaby-Tavor *et al.*, 2020; Bayer; Kaufhold; Reuter, 2022), como observado nas estratégias de cotreinamento.

Para garantir a qualidade semântica da geração neural, a estratégia LAMBADA (*L*anguage-*M*odel-*B*Ased *D*ata *A*ugmentation) (Anaby-Tavor *et al.*, 2020) introduziu a dinâmica de cotreinamento. O método baseia-se no ajuste fino de um modelo gerador (GPT-2) acoplado a um classificador, ambos treinados sobre a base original. Nesse arranjo, o gerador produz novas instâncias textuais de forma condicionada pela classe,

enquanto o classificador atua como um mecanismo de filtragem, selecionando os exemplos com maior probabilidade de pertencer à classe de interesse. Em *benchmarks* multiclasse, a arquitetura demonstrou superioridade preditiva em relação a técnicas como EDA e CBERT, validando a importância do controle de qualidade na geração de dados. Todavia, a metodologia foi restrita a cenários de escassez artificial de dados, nos quais as classes apresentavam distribuições uniformes, deixando inexplorado o comportamento real do método diante do desbalanceamento de classes.

Inspirado nessa mesma dinâmica de cotreinamento, o *framework* PREDATOR (*PRE-trained DATA augmenTOR*) (Abonizio; Junior, 2020) aprimora o conceito ao utilizar modelos mais eficientes: um modelo de linguagem autorregressivo como gerador (DistilGPT2 (Sanh *et al.*, 2019)¹) e um modelo discriminativo como filtro (DistilBERT (Sanh *et al.*, 2019)). Diferentemente do LAMBADA, o modelo gerador usa um conjunto de N exemplos como semente para gerar exemplos sintéticos, filtrados por meio de um limiar na probabilidade de um exemplo pertencer à classe de interesse. Outra diferença-chave é a aplicação do método de forma iterativa, gerando e filtrando exemplos sintéticos até atingir o número desejado. Ao ajustar ambas as arquiteturas à base original, o PREDATOR consegue capturar as nuances específicas do domínio textual. Nos testes conduzidos pelo artigo original, o *framework* superou as demais técnicas. Contudo, assim como no LAMBADA, as avaliações originais concentraram-se estritamente na tratativa de escassez geral de dados em cenários rotulados de forma equilibrada. No entanto, em um trabalho subsequente, esse método é validado em cenários de desbalanceamento (Abonizio; Paraiso; Barbon, 2022).

3.2 Métodos de Sobreamostragem Textual

Esta seção detalha os trabalhos de sobreamostragem textual aplicados para mitigar o desbalanceamento de classes para classificação textual. Conforme detalhado no Capítulo 2, o objetivo da sobreamostragem é gerar exemplos da classe minoritária para mitigar o problema de desequilíbrio de classes. Para mitigar o problema de desbalanceamento de classes na classificação de texto, as abordagens tradicionais recorrem à intervenção nos dados por meio de técnicas de reamostragem. Dentre elas, o algoritmo SMOTE (*Synthetic Minority Over-Sampling Technique*) (Chawla *et al.*, 2002) consolidou-se como um marco metodológico para dados representados em espaços vetoriais contínuos. O método opera gerando novas instâncias da classe minoritária por meio da interpolação linear entre os

¹<https://huggingface.co/distilbert/distilgpt2>

k exemplos mais próximos geometricamente (*k-nearest neighbors*). Contudo, a aplicação direta dessa técnica em PLN exige que a operação ocorra sobre *embeddings* ou representações numéricas de texto. Como consequência de sua natureza puramente geométrica, o SMOTE sintetiza vetores numéricos úteis para a separação das classes, porém desassociados de qualquer estrutura linguística, o que dificulta a geração de novas sequências textuais interpretáveis ou sintaticamente estruturadas na superfície do documento.

Diante dessa limitação, trabalhos recentes passaram a considerar as técnicas de TDA generativas como ferramentas ideais para a sobreamostragem textual coerente. [Abonizio, Paraíso e Barbon \(2022\)](#) apresentaram um estudo comparativo abrangente, com foco em diferentes cenários de classificação, incluindo explicitamente as condições de desbalanceamento de classes. Resgatando os métodos apresentados na Seção 3.1, os autores avaliaram o comportamento de abordagens como EDA, retrotradução e PREDATOR aplicadas a bases de dados de análise de sentimento. As constatações revelaram que, embora tanto o PREDATOR quanto a retrotradução tenham apresentado melhorias significativas diante dos cenários de desbalanceamento, o PREDATOR consolidou-se como o método superior entre as abordagens avaliadas. Por outro lado, a EDA não apresentou melhorias significativas no desempenho, o que reduziu a capacidade preditiva do classificador em determinados cenários. Uma observação crítica levantada pelo estudo, entretanto, é que modelos de classificação modernos baseados em *encoders* já possuem uma robustez intrínseca considerável ao desbalanceamento. Por esse motivo, os métodos de aumento trazem, muitas vezes, apenas melhorias marginais. Além disso, o estudo limitou-se ao domínio de análise de sentimentos.

Seguindo a vertente de métodos baseados em LLMs, [Patil et al. \(Patil et al., 2022\)](#) propuseram o T5W, um método híbrido que integra a capacidade de reescrita do modelo T5 ([Raffel et al., 2020](#)), instruído para a tarefa de geração de paráfrases, ao repositório lexical da base WordNet. Em seus experimentos, os autores avaliaram o impacto do método no treinamento de múltiplos classificadores tradicionais. Os resultados evidenciaram que o T5W não apenas melhora a acurácia global dos modelos, como também supera técnicas consolidadas, tais como a duplicação aleatória (ROS, do inglês Random Over-Sampling), o SMOTE e o EDA. Apesar do rigor comparativo, a eficácia da abordagem foi validada empiricamente em apenas uma base de dados de análise de sentimento (SemEval 2017 ([Rosenthal; Farra; Nakov, 2017](#))).

Focado em Modelos de Linguagem de Grande Porte, [Radliński, Guściora e Kocoń \(2025\)](#) investigaram o impacto prático do aumento de dados gerado por LLMs em contra-

ponto ao uso direto desses modelos como classificadores (*zero-shot* e *few-shot*). Utilizando uma base de dados de análise de sentimentos, o estudo contrastou métodos tradicionais (como SMOTE e retrotradução) com a geração avançada de paráfrases com GPT-3.5 e GPT-4.0. Os experimentos comprovaram que a inserção de dados sintéticos proporcionou melhorias expressivas no desempenho de classificadores baseados em *encoders*, especialmente nas categorias severamente sub-representadas. O trabalho destacou ainda que a técnica de paráfrase por meio de LLM apresentou a maior diversidade léxica entre todas as alternativas mapeadas. Apesar de validar o potencial dos LLMs como geradores de exemplos para sobreamostragem, a investigação limitou-se aos resultados de uma única base de dados.

3.3 Conclusões

A Tabela 1 resume as principais abordagens mapeadas ao longo deste capítulo, contrastando suas dinâmicas de funcionamento, arquiteturas subjacentes, vantagens estruturais e limitações empíricas.

Em síntese, embora a literatura recente consolide a eficácia das técnicas de TDA, as evidências empíricas de métodos generativos baseados em LLMs permanecem frequentemente restritas a avaliações em uma única base de dados (Patil *et al.*, 2022; Radliński; Guściora; Kocoń, 2025) ou a domínios específicos, como a análise de sentimentos (Abonizio; Paraiso; Barbon, 2022).

A presente dissertação busca explorar exatamente essas lacunas. Para tal, ela estabelece uma avaliação comparativa robusta entre métodos de TDA e abordagens clássicas de sobreamostragem e, posteriormente, introduz e valida o método APS. A dissertação traz contribuições à literatura ao conduzir uma análise empírica em múltiplos domínios textuais, demonstrando como a integração de estratégias heurísticas de seleção de instâncias permite refinar o ecossistema de paráfrases, superando de forma consistente tanto a abordagem generativa livre quanto os métodos tradicionais da literatura.

Tabela 1: Comparativo de trabalhos relacionados em sobreamostragem e TDA.

Método / Trabalho	Método de Geração	Método de Seleção	Vantagens	Limitações
SMOTE (Chawla et al., 2002)	Interpolação linear	Nenhum	Baixo custo computacional; eficiente para modelos lineares.	Inviabiliza texto legível; restrito a <i>embeddings</i> .
EDA (Wei; Zou, 2019)	Perturbações heurísticas locais em nível de palavra	Nenhum	Simplicidade algorítmica; dispensa modelos de rede neural.	Ignora estrutura sintática.
CBERT (Wu et al., 2019)	Substituição condicional de <i>tokens</i> via <i>Masked LM</i> (BERT)	Nenhum	Geração local guiada pelo contexto e rótulo.	Inexplorado em cenários de desbalanceamento.
Retrotradução (Edunov et al., 2018)	Rescrita global de sentenças via modelos de tradução	Nenhum	Modifica a sintaxe global; preserva o núcleo semântico.	Eficiência validada apenas em análise de sentimento.
T5W (Patil et al., 2022)	Paráfrase via arquitetura híbrida (T5 + WordNet)	Nenhum	Supera abordagens tradicionais (ROS e EDA).	Validado em apenas uma base de dados (<i>SemEval 2017</i>).
LAMBADA (Anaby-Tavor et al., 2020)	Geração condicional via modelo autorregressivo (GPT-2)	Filtro qualitativo por classificador ajustado	Garante integridade semântica dos sintéticos.	Inexplorado em cenários de desbalanceamento.
PREDATOR (Abonizio; Junior, 2020; Abonizio; Paraíso; Barbon, 2022)	Geração <i>few-shot</i> iterativa autorregressiva (DistilGPT2)	Limiar de confiança τ por classificador (DistilBERT)	Arquitetura leve ($\leq 1B$ de parâmetros); validado sob desbalanceamento real.	Validação restrita a análise de sentimento.
Paráfrase (Radlinski; Gasciera; Kocoń, 2025)	Paráfrase livre contextual por LLMs (GPT-3.5 / GPT-4)	Nenhum	Alta diversidade lexical; forte ganho preditivo.	Alto custo computacional; testado em uma única base.

4 Avaliação de Técnicas de Sobreamostragem Textual

Este capítulo apresenta uma avaliação comparativa entre técnicas clássicas de sobreamostragem e métodos de TDA estabelecidos, incluindo um método baseado em paráfrases, motivado pelo resultado de [Radliński, Guściora e Kocoń \(2025\)](#) que indica que o pré-treinamento de LLMs pode ser suficiente para a geração de exemplos de alta qualidade. As Seções 4.1 a 4.3 detalham a metodologia dos experimentos realizados. Em seguida, a Seção 4.4 apresenta os resultados obtidos, que, por fim, são analisados e discutidos na Seção 4.5, servindo como motivação e justificativa para o desenvolvimento do método proposto nesta dissertação.

4.1 Bases de Dados

Com o objetivo de avaliar cenários distintos de desbalanceamento, as bases de dados escolhidas para essa avaliação abrangem classificação de textos curtos e longos, oriundos de diferentes domínios. As bases de dados foram separadas em dois blocos: i) bases artificialmente desbalanceadas e ii) bases com desbalanceamento natural ou decorrente da binarização. Algumas dessas bases foram originalmente rotuladas como tarefas de classificação multiclasse ou multirrotulo e, quando necessário, foi aplicado um processo de binarização, detalhado em sua descrição. A Tabela 2 resume as bases escolhidas para a avaliação, o número de exemplos em cada classe, a razão de desbalanceamento (RD) da base e a média e desvio padrão da contagem de caracteres.

4.1.1 Bases Artificialmente Desbalanceadas

As bases artificialmente desbalanceadas foram construídas para viabilizar uma avaliação controlada das técnicas investigadas, permitindo isolar os efeitos do desbalanceamento de classes sobre o desempenho dos classificadores. Além de sustentarem as análises desta dissertação, essas bases poderão servir de cenários de referência para estudos futuros

Tabela 2: Resumo das estatísticas das bases de dados avaliadas.

Base	Exemplos de Classe 0	Exemplos de Classe 1	RD	Tamanho médio de sentença	Desvio padrão de sentença
CI/SST-2	1.186	37.330	31,48	52,31	43,26
CI/SemEval 2017	129	7.046	54,62	112,62	26,51
CI/IMDB	497	12.472	25,09	1.344,95	1.043,00
CI/Rotten Tomatoes	426	4.265	10,01	114,51	51,27
SemEval 2015	274	47	5,83	97,86	29,02
SemEval 2024	1.264	5.736	4,54	116,10	106,09
SemEval 2025	2.432	332	7,32	78,41	54,79
Twitter Topics (0 vs All)	6.844	152	45,03	166,79	68,29
Twitter Topics (1 vs All)	6.679	317	21,07	166,79	68,29

sobre balanceamento de classes. Essas bases foram criadas a partir de bases sem desbalanceamento significativo, submetidas a um processo de subamostragem controlada. O procedimento consistiu na remoção aleatória de instâncias da classe minoritária, mantendo 50%, 25%, 10%, 5%, 4% e 1% do volume original desse subconjunto. Cada um dos seis níveis de subamostragem foi realizado cinco vezes, com sementes de aleatoriedade 1, 2, 3, 4 e 5, obtendo-se uma média de desempenho (F1-Score) para cada nível.

Para medir o desempenho da base, adotou-se o modelo de linguagem *XLM-RoBERTa* (XLM-R) (Conneau *et al.*, 2020), acoplado a uma camada de classificação padrão da arquitetura (*sequence classification head*). A escolha do modelo XLM-RoBERTa fundamenta-se na presença de termos e sentenças em múltiplos idiomas nas bases avaliadas, bem como em sua natureza *open-source* e no desempenho competitivo reportado em estudos anteriores (Cunha; Rocha; Gonçalves, 2025). O processo de treinamento consistiu no ajuste fino de todos os parâmetros do modelo por até 30 épocas, utilizando a técnica de parada antecipada com paciência de até três épocas. Essa escolha segue o padrão de uma estratégia de classificação agnóstica à base de dados (Fonseca *et al.*, 2025). A validação intermediária a cada época foi realizada em um subconjunto de 10% do conjunto de treino. Para avaliar o desempenho preditivo do modelo treinado, foram previamente separados 20% dos exemplos da base para o conjunto de teste.

Considerando-se o desempenho dos seis níveis de desbalanceamento, de 50% a 1% nessa ordem, selecionou-se o desbalanceamento mais severo imediatamente anterior ao colapso completo do modelo (isto é, quando o desempenho do modelo, medido pelo F1-Score, cai a zero). Dentre as cinco repetições para cada nível de desbalanceamento, foi selecionada a partição que apresentou o melhor desempenho, buscando preservar exemplos mais representativos da classe minoritária e, conseqüentemente, reduzir o risco de remover instâncias relevantes (ou representativas) da sua distribuição original.

A Figura 1 ilustra os resultados desse processo nas quatro bases, com destaque para a razão de exemplos mantidos adotada; as bases resultantes do processo de desbalanceamento recebem um prefixo **CI/** em sua denominação, que se refere à expressão Desbalanceamento de Classes (do inglês, *Class Imbalance*).

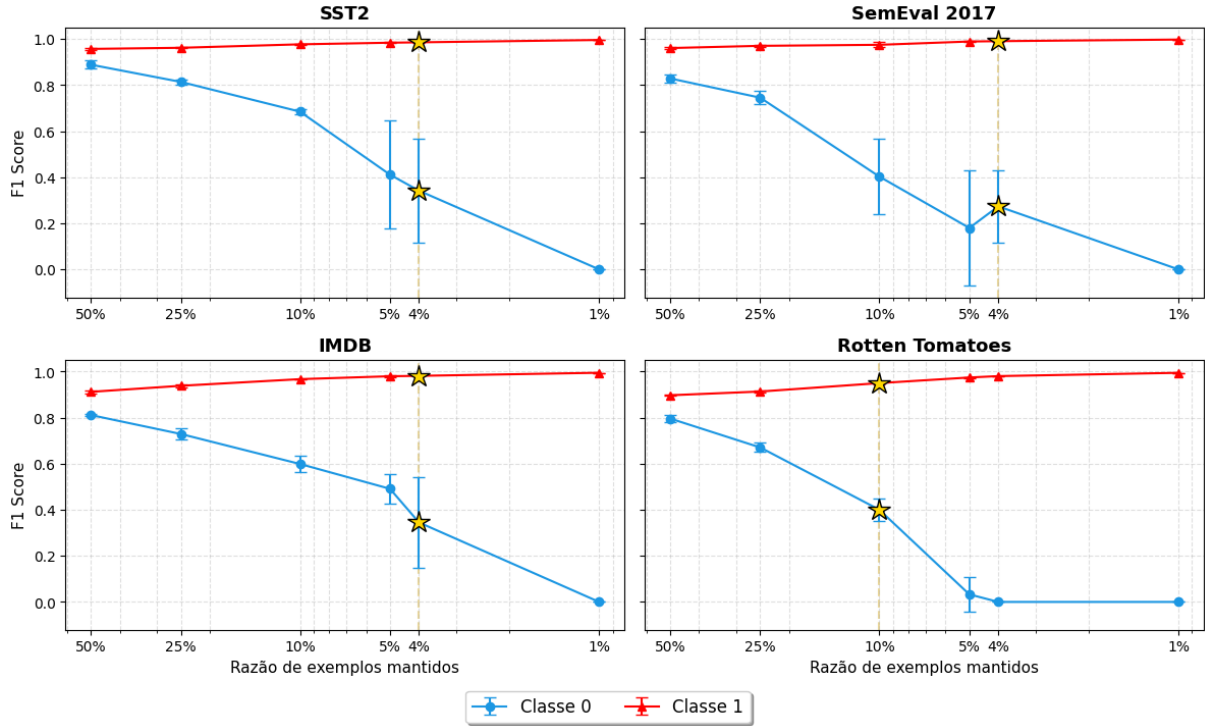


Figura 1: Resultado do processo de desbalanceamento artificial. A razão escolhida para cada base foi destacada por uma estrela.

A base SST-2 (Socher *et al.*, 2013) fornece sentenças curtas extraídas de críticas de cinema do *benchmark* GLUE, captando nuances semânticas em estruturas concisas. A base resultante do processo de desbalanceamento, CI/SST-2, mantém 4% dos exemplos da classe minoritária. Em contraste, a IMDB Movie Reviews (Maas *et al.*, 2011) apresenta textos longos e multifrasais, o que demanda a modelagem de dependências de longo alcance. A base resultante, denominada CI/IMDB, mantém 4% dos exemplos minoritários. A base Rotten Tomatoes (Pang; Lee, 2005) atua como um contraponto ao IMDB, oferecendo resenhas condensadas de críticos profissionais. A base desbalanceada, CI/Rotten Tomatoes mantém 10% dos exemplos minoritários. Por fim, no contexto das redes sociais, a base SemEval 2017 (Task 4) (Rosenthal; Farra; Nakov, 2017) introduz desafios de robustez decorrentes do ruído linguístico típico do Twitter (atual X), como abreviações e *emojis*. Diferentemente das demais, essa base categoriza seus exemplos como positivos, neutros e negativos e sua binarização foi realizada por meio da exclusão completa das instâncias neutras. Após essa exclusão, a base foi submetida ao mesmo processo de des-

balanceamento, resultando na base CI/SemEval 2017, na qual 4% dos exemplos da classe minoritária foram mantidos.

4.1.2 Bases com Desbalanceamento Natural ou Decorrente da Binarização

O segundo bloco é composto por bases que não passaram por manipulação artificial da distribuição de classes, exibindo desbalanceamento nativo ou gerado pelo processo de binarização no caso de bases multiclasse. A base SemEval 2015 (Task 11) (Ghosh *et al.*, 2015) aborda a análise de sentimentos em linguagem figurada na língua inglesa no Twitter. Devido às restrições de expiração da interface de programação de aplicações (API) da plataforma, coletou-se um subconjunto amostral de 274 exemplos negativos (0) e 47 exemplos positivos (1). A base SemEval 2024 (Task 4) (Dimitrov *et al.*, 2024) aborda a detecção e classificação de técnicas de persuasão em *memes*, em inglês. Para este trabalho, as classes originais, compostas por uma lista de técnicas de persuasão presentes em cada exemplo, foram binarizadas para indicar apenas a presença de pelo menos uma técnica (positivo) ou a ausência de qualquer técnica (negativo). Para cenários multilíngues e de múltiplas fontes, utilizou-se a base de dados SemEval 2025 (Task 11) (Muhammad *et al.*, 2025). Como cada texto pode ser rotulado com múltiplos sentimentos simultâneos, aplicou-se a estratégia de Relevância Binária para isolar uma única emoção que manifestasse um grau elevado de assimetria nativa, mapeando como 1 a presença do sentimento e 0 a sua ausência (incluindo textos neutros). Dentre as emoções presentes na base, foi isolada a emoção de raiva, que apresenta um desbalanceamento elevado ($RD = 7,32$). Por fim, a base *Twitter Topic Classification* (Antypas *et al.*, 2022) introduz o contexto de tópicos gerais nas redes sociais por meio de seis categorias. Para adequá-la ao escopo binário, aplicou-se a estratégia *One-versus-All* para derivar duas bases independentes: 0 vs All, que isola a classe Arte e Cultura (0), e 1 vs All, que isola a classe Negócios e Empreendedorismo (1).

4.2 Métodos de Aumento de Dados Avaliados

No estudo conduzido neste capítulo, serão exploradas quatro estratégias de aumento de dados que representam diferentes paradigmas da literatura. Todos os métodos avaliados foram configurados para atingir o balanceamento entre as classes ($\alpha = 1,0$). Vale ressaltar que nem todos os métodos alcançam perfeitamente $\alpha = 1,0$, o que gera um número de exemplos para o qual $\alpha \approx 1,0$. Os métodos avaliados são detalhados nas subseções a

seguir.

4.2.1 SMOTE

O algoritmo SMOTE (Chawla *et al.*, 2002) realiza o rebalanceamento operando diretamente no espaço de características no qual os textos foram vetorizados. O processo inicia-se com a seleção de uma instância semente x_i do conjunto minoritário D_{min} . Em seguida, calcula-se a distância euclidiana entre x_i e todos os demais exemplos de D_{min} para identificar seus k vizinhos mais próximos. Um desses vizinhos, representado por x_j , é selecionado aleatoriamente para orientar a criação do novo exemplo. A nova instância sintética x_{new} é gerada por meio da interpolação linear entre a semente x_i e o vizinho escolhido x_j , conforme

$$x_{new} = x_i + \lambda(x_j - x_i) \quad (4.1)$$

onde λ é um número aleatório extraído de uma distribuição uniforme no intervalo $[0, 1]$. Esse mecanismo garante que a nova representação numérica seja posicionada ao longo do segmento de reta que une os dois exemplos originais. A implementação usada desse método foi disponibilizada pela biblioteca *imbalanced-learn*¹.

4.2.2 EDA

Diferentemente das abordagens operadas no espaço vetorial, o método EDA (Wei; Zou, 2019) atua diretamente na superfície do texto, aplicando perturbações heurísticas locais em nível de palavra. Para cada exemplo semente selecionado em D_{min} , o operador do EDA cria k (parâmetro especificado pelo usuário) exemplos sintéticos, de forma aleatória, a partir de uma das quatro operações discretas:

- **Substituição por Sinônimos (SR):** Seleciona n palavras aleatórias do exemplo (que não sejam palavras funcionais, *stopwords*) e as substitui por sinônimos extraídos de uma base lexical como a WordNet.
- **Inserção Aleatória (RI):** Escolhe aleatoriamente um sinônimo de uma palavra presente no exemplo e insere-o em uma posição arbitrária (definida aleatoriamente) do texto. Esse processo é repetido n vezes.

¹<https://imbalanced-learn.org/>

- **Troca Aleatória (RS):** Escolhe duas palavras aleatoriamente no exemplo e as troca de posição. Essa permutação é executada n vezes.
- **Deleção Aleatória (RD):** Para cada palavra do exemplo, o operador tem probabilidade p de remover a palavra atual.

Os parâmetros n das operações discretas são calculados multiplicando o número de palavras do exemplo semente pelos parâmetros de cada operação: α_{SR} , α_{RI} e α_{RS} . E a probabilidade p é especificada diretamente como um parâmetro. Estes parâmetros foram mantidos no valor padrão de 0,1. A implementação desse método foi disponibilizada pelo artigo em um repositório².

4.2.3 PREDATOR

O PREDATOR (Abonizio; Junior, 2020) propõe uma abordagem de aumento de dados baseada em um processo de geração e filtragem. O método utiliza dois componentes principais estruturados de forma independente: um modelo de linguagem autorregressivo (gerador) e um classificador discriminativo (filtro). Ambos os modelos são ajustados com o conjunto de dados de treinamento original para captar as nuances e o estilo linguístico do domínio. A Figura 2 ilustra a arquitetura geral do PREDATOR.

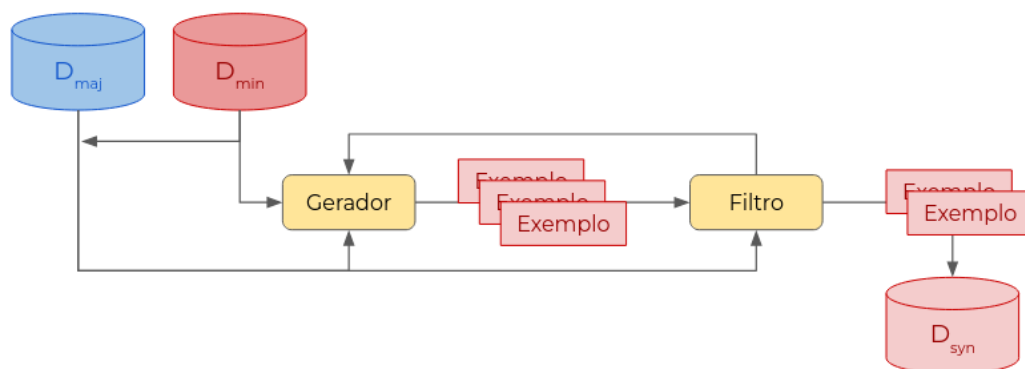


Figura 2: Arquitetura geral do PREDATOR.

O processo inicia-se com o gerador produzindo novas sequências textuais com base em uma estratégia de aprendizado em contexto (*in-context learning*) (Brown *et al.*, 2020). O *prompt* fornecido ao gerador consiste em N exemplos selecionados aleatoriamente de D_{min} (cujo valor padrão, estabelecido na literatura, é 3), concatenados por meio do *token* de separação nativo do modelo. Em seguida, cada sentença sintética gerada, denotada

²https://github.com/jasonwei20/eda_nlp/tree/master

por $x_{new} \in D_{gen}$, é submetida à inferência do modelo discriminativo. O filtro calcula a probabilidade condicional da instância pertencer à classe minoritária de interesse. O exemplo sintético x_{new} é retido e adicionado ao conjunto D_{syn} se, e somente se, satisfizer o seguinte critério de corte:

$$P(y_i | x_{new}) \geq \tau \quad (4.2)$$

onde y_i é uma classe pertencente ao conjunto discreto de classes Y e $\tau \in [0, 1]$ é um hiperparâmetro de limiar de confiança, cujo valor padrão estabelecido na literatura é 0,75. Caso a probabilidade seja inferior a τ , a instância é descartada. Os hiperparâmetros N e τ , assim como os empregados no ajuste dos modelos gerador e filtro, foram definidos de acordo com os valores usualmente adotados na literatura (Abonizio; Junior, 2020; Abonizio; Paraiso; Barbon, 2022). Da mesma forma, foram utilizados os modelos de referência DistilGPT2, como gerador, e DistilBERT, como filtro. A implementação desse método foi disponibilizada no artigo original, em um repositório³.

4.2.4 Paráfrases Geradas por LLMs

O aumento de dados baseado em paráfrases geradas por LLMs representa uma evolução em relação aos métodos de geração tradicionais. Em vez de treinar ou ajustar um modelo gerador específico para o domínio, essa abordagem explora a capacidade de generalização e o conhecimento prévio de LLMs. Foi empregado o modelo *meta-llama/Llama-3.1-8B-Instruct* (Grattafiori et al., 2024), considerando a capacidade do ambiente computacional e sua natureza *open-source*. A implementação foi feita por meio da biblioteca *Hugging-Face* (Wolf et al., 2020). A Figura 3 ilustra a arquitetura geral do método.

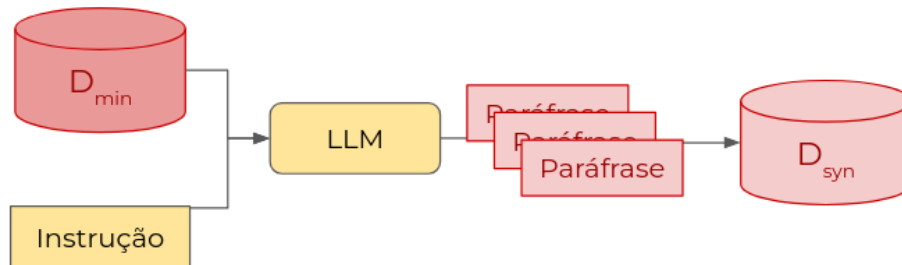


Figura 3: Arquitetura geral do método de paráfrases geradas por LLMs.

A instrução do modelo (*prompt*) é estruturada por meio de técnicas de engenharia de

³<https://github.com/hugoabonizio/predator>

prompt e *meta-prompting*. Esse processo consiste em refinar as instruções textuais fornecidas ao modelo iterativamente. O objetivo do *meta-prompting*, neste contexto, é mitigar alucinações semânticas, preservar o sentido original da sentença-semente e assegurar o cumprimento de formatos de saída padronizados (como estruturas em JSON), facilitando a extração automatizada de textos sintéticos. O Quadro 1 detalha o *prompt*; um texto entre chaves duplas (`{{}}`) indica que a informação é um parâmetro inserido pelo usuário.

Quadro 1: Instrução de sistema utilizado para geração de paráfrases.

Instrução de sistema

SYSTEM ROLE

You are a High-Fidelity Style Transfer & Data Augmentation Engine. Your goal is to generate $N={{N}}$ paraphrases while aggressively preserving the specific "voice" of the input text.

CRITICAL: STYLE MIRRORING

The most common error is "normalizing" text. You must NOT do this.

1. **Analyze the Input:** Look for capitalization patterns (e.g., all lowercase), punctuation habits (e.g., multiple !!!), slang, abbreviations (e.g., "u" vs "you"), and emojis.
2. **Mimic Strictly:** If the input is a messy Tweet, the output MUST be a messy Tweet. If the input is formal code documentation, the output MUST be formal.
3. **Do Not Correct:** Do not fix grammar, do not expand abbreviations, and do not formalize the tone.
4. **Language Anchor:** You must generate the output in the exact same language as the input. Even if the input is in Brazilian Portuguese, the output must remain in Brazilian Portuguese. Do not translate or localize.

CRITICAL: OUTPUT FORMAT & TOKEN LIMITS

You are interacting with a parser that crashes on invalid JSON.

1. **Raw JSON Only:** Do not use markdown blocks (``json). Do not add introductions or conclusions.
2. **Atomic Batches:** Output a valid JSON object containing a list. If you feel you are approaching a token limit, close the JSON list and object early rather than cutting off in the middle of a string.
3. **Structure:** `{"paraphrases": ["paraphrase_1", "paraphrase_2"]}`

FEW-SHOT EXAMPLES (Style Fidelity)

Example 1: Informal/Tweet Style

Input: "omg i cant believe he said that lol 🤔"

N: 1

Result:

`{"paraphrases": ["lol i am literally in shock that he said that 🤔"]}`

Example 2: Technical/Formal Style

Input: "The function aborts if the return value is null."

N: 1

Result:

`{"paraphrases": ["If a null value is returned, the procedure terminates."]}`

Para cada exemplo original extraído de D_{min} , o modelo é instruído a gerar cinco paráfrases (pelo parâmetro N do *prompt*) de forma iterativa até atingir uma quantidade k de paráfrases por instância-semente – essa limitação é imposta para preservar o limite de contexto do modelo. Para estimular a diversidade das saídas entre as iterações, adotou-se temperatura de 1,0. Os demais parâmetros de amostragem foram mantidos nos padrões originais do modelo, incluindo o *top-p* (0,9). Para garantir uma distribuição homogênea no rebalanceamento, o fator de multiplicação k é definido de maneira igualitária para todos os exemplos da classe minoritária, sendo delimitado pelo maior inteiro que satisfaça a restrição de teto do conjunto majoritário:

$$k = \max \left\{ k \in \mathbb{Z} \mid (k + 1) \times |D_{min}| \leq |D_{maj}| \right\} \quad (4.3)$$

Essa estratégia assume que todas as instâncias pertencentes a D_{min} possuem igual relevância estatística para o classificador, expandindo o espaço amostral da classe minoritária uniformemente antes de aplicar qualquer critério de seleção ou descarte de dados.

4.3 Estratégias de Classificação

Para a tarefa de classificação, foram avaliadas duas estratégias sobre representações geradas pelo XLM-RoBERTa (XLM-R) (Conneau *et al.*, 2020). A escolha do modelo XLM-RoBERTa fundamenta-se na presença de termos e sentenças em múltiplos idiomas nas bases avaliadas, bem como em sua natureza *open-source* e no desempenho competitivo reportado em estudos anteriores (Cunha; Rocha; Gonçalves, 2025). Em ambas as estratégias, o modelo XLM-R foi implementado por meio da biblioteca HuggingFace (Wolf *et al.*, 2020).

Na primeira estratégia, o classificador baseado em regressão logística é treinado com os *embeddings* extraídos por meio de *mean-pooling* (no nível de *token*, a partir dos *embeddings* da última camada) do modelo XLM-R. O objetivo dessa estratégia é isolar a qualidade das representações latentes e avaliar o desempenho do SMOTE em comparação com os métodos de TDA. O modelo de regressão logística foi implementado utilizando a biblioteca *SciKit* (Pedregosa *et al.*, 2011), com até 1.000 iterações de treino e os demais hiperparâmetros padrão da implementação. A escolha de um modelo de regressão logística se justifica na sua semelhança à camada de classificação (*sequence classification head*) tradicionalmente implementados para adaptar a saída de modelos de linguagem baseados em *encoder* para a tarefa da classificação de textos. Na segunda estratégia, utilizou-se

o próprio XLM-R como classificador, acoplado a uma camada de classificação padrão da arquitetura. O processo de treinamento consistiu no ajuste fino de todos os parâmetros do modelo por até 30 épocas, utilizando a técnica de parada antecipada com paciência de três épocas. A validação intermediária a cada época foi realizada em um subconjunto de 10% dos dados de treinamento separados automaticamente. Essa escolha segue o padrão de uma estratégia de classificação agnóstica à base de dados (Fonseca *et al.*, 2025).

Com o intuito de garantir um comparativo robusto e compreender a capacidade de generalização das abordagens, todas as avaliações foram conduzidas segundo o protocolo de validação cruzada com 10 partições. Para avaliar o desempenho preditivo dos modelos em cenários de desbalanceamento, adotou-se o *F1-Score* como métrica principal (He; Garcia, 2009). Como o *F1-Score* tradicional é calculado tendo uma classe específica como referência, adota-se a variante *Macro-F1* para a agregação dos resultados. Essa abordagem garante que o desempenho na classe minoritária tenha o mesmo peso estatístico que o da classe majoritária, impedindo que a acurácia global mascare eventuais colapsos do modelo.

4.4 Resultados

As Tabelas 3 e 4 detalham os resultados de *Macro-F1* (média $\pm$ desvio padrão entre os 10 *folds*) no protocolo de validação cruzada, nas nove bases testadas. Números em negrito indicam o melhor desempenho médio da base. Já números sublinhados indicam significância estatística em relação ao cenário sem balanceamento (primeira coluna) por meio do teste de *Wilcoxon Rank-Sum* (Wilcoxon, 1945), com $\alpha = 0,05$. Também é reportada a posição média no *ranking* de resultados (ou seja, o desempenho relativo entre os métodos, considerando todas as bases avaliadas, em que valores menores indicam posições melhores).

Tabela 3: Resultados da Regressão Logística para as nove bases.

Base	Sem Balanceamento	SMOTE	EDA	PREDATOR	Paráfrase
CI/SST-2	0,5257 $\pm$ 0,0133	0,5773 $\pm$ 0,0071	0,5656 $\pm$ 0,0095	0,5412 $\pm$ 0,0074	0,5744 $\pm$ 0,0071
CI/SemEval 2017	0,5170 $\pm$ 0,0346	0,6182 $\pm$ 0,0274	0,5757 $\pm$ 0,0559	0,5722 $\pm$ 0,0623	0,6037 $\pm$ 0,0228
CI/IMDB	0,5099 $\pm$ 0,0091	0,6165 $\pm$ 0,0151	0,5521 $\pm$ 0,0473	0,6146 $\pm$ 0,0371	0,6114 $\pm$ 0,0192
CI/Rotten Tomatoes	0,5110 $\pm$ 0,0251	0,6544 $\pm$ 0,0225	0,5966 $\pm$ 0,0369	0,6131 $\pm$ 0,0353	0,5995 $\pm$ 0,0392
SemEval 2015	0,5195 $\pm$ 0,0982	0,7731 $\pm$ 0,1185	0,6433 $\pm$ 0,1634	0,6983 $\pm$ 0,1548	0,6539 $\pm$ 0,1276
SemEval 2024	0,6658 $\pm$ 0,0250	0,7045 $\pm$ 0,0174	0,6752 $\pm$ 0,0259	0,7084 $\pm$ 0,0224	0,6751 $\pm$ 0,0246
SemEval 2025	0,5241 $\pm$ 0,0228	0,6588 $\pm$ 0,0274	0,5970 $\pm$ 0,0646	0,6693 $\pm$ 0,0450	0,6218 $\pm$ 0,0391
Twitter Topics (0 vs All)	0,5008 $\pm$ 0,0199	0,6522 $\pm$ 0,0193	0,5478 $\pm$ 0,0670	0,6641 $\pm$ 0,0810	0,6335 $\pm$ 0,0491
Twitter Topics (1 vs All)	0,7350 $\pm$ 0,0486	0,7608 $\pm$ 0,0186	0,7210 $\pm$ 0,0448	0,7850 $\pm$ 0,0432	0,7365 $\pm$ 0,0422
Ranking Médio	4,9	1,4	3,8	2,0	2,9

Os resultados mostram que o impacto das técnicas de balanceamento varia conforme o

Tabela 4: Resultados do *fine-tuning* do XLM-R para as nove bases.

Base	Sem Balanceamento	EDA	PREDATOR	Paráfrase
CI/SST-2	$0,7013 \pm 0,0476$	$0,7384 \pm 0,0320$	$0,6703 \pm 0,0238$	$0,7313 \pm 0,0190$
CI/SemEval 2017	$0,5865 \pm 0,0961$	$0,6486 \pm 0,0685$	$0,6220 \pm 0,0594$	$0,6968 \pm 0,0620$
CI/IMDB	$0,6785 \pm 0,0880$	$0,6639 \pm 0,1040$	$0,6690 \pm 0,0656$	$0,6721 \pm 0,0450$
CI/Rotten Tomatoes	$0,6255 \pm 0,1155$	$0,5209 \pm 0,0599$	$0,6398 \pm 0,0891$	$0,5755 \pm 0,0930$
SemEval 2015	$0,4792 \pm 0,0582$	$0,6772 \pm 0,1641$	$0,6541 \pm 0,1620$	$0,6343 \pm 0,1571$
SemEval 2024	$0,7030 \pm 0,0264$	$0,6877 \pm 0,0180$	$0,7026 \pm 0,0254$	$0,6600 \pm 0,0315$
SemEval 2025	$0,6343 \pm 0,0948$	$0,5607 \pm 0,0854$	$0,6632 \pm 0,0314$	$0,6146 \pm 0,0437$
Twitter Topics (0 vs All)	$0,6087 \pm 0,1175$	$0,6095 \pm 0,1048$	$0,6697 \pm 0,0829$	$0,6651 \pm 0,0700$
Twitter Topics (1 vs All)	$0,8201 \pm 0,0156$	$0,8127 \pm 0,0140$	$0,8116 \pm 0,0349$	$0,7810 \pm 0,0972$
Ranking Médio	2,4	2,7	2,2	2,7

classificador utilizado. Como observado na Tabela 3, o classificador baseado em Regressão Logística (LR) se beneficiou de todos os métodos, alcançando melhorias estatisticamente significativas no desempenho em todas as bases de dados. Entre os métodos aplicados a esse classificador tradicional, o SMOTE se destacou e obteve a melhor posição no *ranking* médio (1,4), seguido pelo PREDATOR (2,0).

Por outro lado, como ilustrado na Tabela 4 o modelo XLM-R já apresentou um desempenho inicial elevado em um subconjunto das bases sem o uso de técnicas de balanceamento, corroborando os resultados de Abonizio, Paraíso e Barbon (Abonizio; Paraíso; Barbon, 2022). Nestes casos, as estratégias de balanceamento não resultaram em melhorias significativas e, em alguns casos, até reduziram o desempenho médio do modelo, levando a perdas em termos de efetividade. Analisando o *ranking* médio, observa-se que os métodos de EDA e paráfrase apresentaram resultados inferiores aos do cenário sem mitigação de balanceamento (2,7). Das nove bases testadas, ambas as estratégias pioraram o desempenho em cinco delas. Observando o *ranking* médio nesses casos, apenas o PREDATOR apresenta posição marginalmente superior à do método sem balanceamento (2,4).

A Tabela 5 indica qual estratégia de balanceamento apresentou o maior desempenho médio em cada base. A primeira coluna considera apenas os métodos combinados à regressão logística, a segunda coluna considera apenas os métodos combinados ao XLM-R e a terceira coluna considera todos os pares classificador-método.

A base CI/SST-2 foi a única base a atingir seu melhor desempenho médio por meio do EDA, acoplado ao XLM-R. Por sua vez, PREDATOR e paráfrase dominaram os cenários de mídias sociais e de tarefas complexas: a base CI/SemEval 2017 obteve seu maior desempenho médio com paráfrase combinada ao XLM-R, enquanto o *framework* PREDATOR demonstrou forte sinergia com classificadores lineares, impulsionando o desempenho no

Tabela 5: Estratégia com melhor desempenho médio por base.

Base de Dados	Melhor Estratégia em LR	Melhor Estratégia em XLM-R	Melhor Estratégia
CI/SST-2	SMOTE	EDA	EDA + XLM-R
CI/SemEval 2017	SMOTE	Paráfrase	Paráfrase + XLM-R
CI/IMDB	SMOTE	Sem Balanceamento	Sem Balanceamento + XLM-R
CI/Rotten Tomatoes	SMOTE	PREDATOR	SMOTE + LR
SemEval 2015	SMOTE	EDA	SMOTE + LR
SemEval 2024	PREDATOR	Sem Balanceamento	PREDATOR + LR
SemEval 2025	PREDATOR	PREDATOR	PREDATOR + LR
Twitter Topics (0 vs All)	PREDATOR	PREDATOR	PREDATOR + XLM-R
Twitter Topics (1 vs All)	PREDATOR	Sem Balanceamento	Sem Balanceamento + XLM-R

SemEval 2024 e no SemEval 2025 quando acoplado à regressão logística. Adicionalmente, o PREDATOR também garantiu o melhor resultado para o *Twitter Topics (0 vs All)*, quando integrado ao XLM-R. Em contrapartida, o algoritmo SMOTE acoplado à regressão logística foi a combinação mais eficaz para as bases de resenhas cinematográficas do CI/Rotten Tomatoes e para o domínio do SemEval 2015. Por fim, os cenários baseados no CI/IMDB e no Twitter Topics (1 vs All) prescindiram de qualquer balanceamento, atingindo seus melhores resultados com o XLM-R operando de forma nativa sobre a distribuição original desbalanceada. Portanto, nenhum método ou estratégia de classificação domina as nove bases, o que evidencia um empate entre os métodos avaliados.

Para confirmar se as diferenças entre os resultados obtidos são significativas, foi feito um teste estatístico de Friedman com $p \leq 0,05$ seguido do teste *post-hoc* Nemenyi. Essa combinação foi escolhida por ser amplamente recomendada para a comparação simultânea de múltiplos algoritmos em diferentes bases de dados (Demšar, 2006). Os resultados da regressão logística demonstraram diferença significativa entre os métodos: Friedman = 25,8667 e $p = 3,3665 \times 10^{-5}$. Por outro lado, os métodos em XLM-R não apresentaram diferenças significativas entre si (Friedman = 0,7333 e $p = 0,8653$).

Analisando os resultados par a par do teste *post-hoc* de Nemenyi entre os métodos acoplados à regressão logística, não foram encontradas diferenças significativas entre SMOTE e PREDATOR ($p = 0,946$). Em suma, o desempenho entre os dois classificadores é contrastante: enquanto a regressão logística se beneficia de todos os métodos, o XLM-R não apresenta diferenças significativas de desempenho.

Um segundo teste de Friedman foi realizado para todos os pares classificador-método, demonstrando diferença significativa entre os pares (Friedman = 32,2963 e $p = 8,2419 \times 10^{-5}$). Avaliando os valores p par a par do teste *post-hoc* de Nemenyi, observa-se que os resultados obtidos com todos os pares classificador-método diferem significativamente da regressão logística sem balanceamento ($0 \leq p \leq 0,039$), com exceção do EDA e paráfrase ($p = 0,972e0,497$, respectivamente).

Por fim, para investigar os fatores que podem ter contribuído para os resultados observados, apresentamos exemplos na Tabela 6, comparando as instâncias geradas aos respectivos exemplos originais. Os exemplos foram selecionados a partir dos cenários vencedores identificados na Tabela 5, considerando apenas os métodos cujas saídas são textos.

Tabela 6: Exemplos gerados pelas técnicas vencedoras em cada base.

Base	TDA Vencedor	Texto Original	Texto Gerado
CI/SST-2	EDA	, you 're far better served by the source material .	you re far better served by the source textile
CI/SemEval 2017	Paráfrase	Chris Evans just said exactly what I was thinking about Saturday Kitchen 'I don't care about this show anymore'	Chris Evans just echoed my sentiments about Saturday Kitchen, I'm so over it now
SemEval2024	PREDATOR	THIS WAS GO FUND ME WHEN I WAS AKID	IT WAS. WHEN I HAD I'LL DO
SemEval2025	PREDATOR	I stomped my foot in anger.	ed my feet in pain, spit on my nose, and then ran away!
Twitter (0 vs All)	PREDATOR	This book provides the perfect quick read Don t miss Truth and Grace Homeschool Academy s review of Ghosted at the Altar by {{USERNAME}} #book & enter the #free #giveaway for a \$25 Amazon gift card! #books #amreading {{URL}}	great book so read up for you). I will be sharing the book with you as I read it on my blog {{URL}} and if you have questions like that, you can

No caso do EDA, observa-se que os exemplos gerados são injetados com ruídos textuais; a substituição de “*material*” por “*textile*” ilustra a limitação do método em gerar perturbações que mantenham a integridade do texto. A saída do PREDATOR é frequentemente truncada, gerando frases incompletas. Em contrapartida, como consequência do ajuste de domínio, o método gera exemplos fiéis ao estilo linguístico⁴, ilustrados pela capitalização e pela quebra de linha em SemEval 2024. Em contraste, a paráfrase por meio de LLMs se vale do pré-treinamento de um modelo mais robusto, criando textos coerentes e moderadamente distintos lexicalmente, preservando a semântica original. O desempenho marginalmente superior do PREDATOR no comparativo de desempenho sugere que o passo de ajuste dos modelos pode contribuir para a generalização do método em bases distintas, adaptando-o às particularidades linguísticas de cada base.

⁴Como o processo de geração do PREDATOR não mantém uma correspondência explícita entre exemplos originais e sintéticos, cada exemplo sintético foi associado ao exemplo de D_{min} com o qual apresentou a maior similaridade de cosseno.

4.5 Discussão

A análise dos resultados deste estudo permite extrair conclusões sobre o impacto das técnicas de aumento de dados. Observa-se que SMOTE e PREDATOR, quando combinadas com o classificador LR, produziram ganhos significativos de desempenho. Contudo, esse comportamento não se repetiu ao utilizar o modelo XLM-R: a incorporação das técnicas de aumento ao XLM-R não resultou em melhorias preditivas consistentes e, em alguns cenários, levou à degradação do desempenho do classificador. Embora o PREDATOR tenha apresentado uma vantagem marginal em termos absolutos, o teste estatístico *post-hoc* não confirmou diferenças significativas entre os métodos avaliados em conjunto com o XLM-R.

A análise qualitativa dos exemplos gerados ajuda a explicar esses resultados. O EDA, por sua natureza baseada em perturbações locais, introduz ruído no texto. O PREDATOR, apesar do desempenho marginalmente superior, frequentemente gera frases truncadas e, por consequência, sem coerência textual. Em contraste, a geração de paráfrases por meio de LLM foi a única técnica capaz de produzir textos sintaticamente coerentes e fluidos.

Com base nessas observações, responde-se diretamente à primeira questão de pesquisa (QP1): a sobreamostragem textual baseada em LLMs não se mostra mais eficaz do que as abordagens tradicionais para mitigar o desbalanceamento de classes.

Em particular, o fato de a abordagem baseada em paráfrase produzir exemplos linguisticamente mais coesos e, ainda assim, não superar de forma consistente as técnicas tradicionais de balanceamento indica que a qualidade linguística, por si só, não é suficiente para garantir a utilidade dos exemplos sintéticos na tarefa final de classificação. A simples geração de paráfrases pode criar instâncias que, embora semanticamente válidas, mostram-se redundantes ou pouco informativas para o ajuste da fronteira de decisão. Esse resultado sugere que a simples geração de novos exemplos não é suficiente; é igualmente importante identificar e filtrar aqueles com maior potencial de contribuir efetivamente para o treinamento. Esses achados justificam a necessidade de mecanismos dedicados de seleção de qualidade, constituindo o alicerce para o método de balanceamento proposto no próximo capítulo.

5 Paráfrases com LLMs e Seletores Heurísticos

Este capítulo detalha o método Aumento por Paráfrases Seleccionadas (APS) proposto nesta dissertação. A Seção 5.1 descreve a arquitetura geral do método. Em seguida, as Seções 5.2 e 5.3 detalham seus componentes principais: a criação de paráfrases por meio de LLMs e o método de seleção de paráfrases.

5.1 Arquitetura Geral do Método

Tendo como motivação os bons resultados alcançados pelo PREDATOR, que é constituído por dois componentes principais, geração e seleção. Ainda, considerando os resultados obtidos por meio da geração de paráfrases observados no Capítulo 4, em que as paráfrases geradas demonstram coerência e fluidez nos exemplos gerados. No entanto, o PREDATOR tem como desvantagem a necessidade de ajustar o modelo gerador, o que limita o uso de LLMs mais robustas em cenários com baixa capacidade computacional (Schwartz *et al.*, 2020). Também, conforme discutido, a geração de paráfrases por si só não gera melhoras significativas de desempenho. Assim, o método proposto é projetado para mitigar o desbalanceamento de classes por meio da geração e da seleção heurística de paráfrases. A Figura 4 ilustra o fluxo de dados e de execução dos principais componentes que integram o método:

1. **Instanciação e Geração de Paráfrases (LLM):** Este bloco é responsável por carregar a LLM local e executar o mecanismo de *prompting*. O modelo recebe cada exemplo pertencente à classe minoritária, contido na base de treinamento, e gera paráfrases candidatas.
2. **Seleção Heurística Pós-Geração (Seletor):** Este componente atua como um filtro qualitativo. Serão consideradas duas vertentes metodológicas para esta triagem: a **Entropia de Shannon**, que prioriza exemplos na fronteira de decisão por meio de um modelo classificador discriminativo probabilístico, denominado *probe*, e

a **Proximidade ao Vetor Protótipo**, que prioriza o alinhamento de domínio no espaço latente. O seletor por entropia ainda inclui um passo opcional de ajuste fino prévio do *probe*.

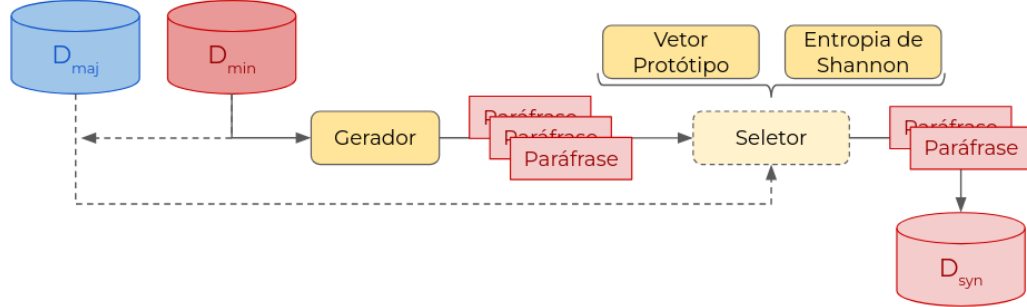


Figura 4: Arquitetura geral do método proposto APS.

Em comparação com abordagens estabelecidas na literatura, como o LAMBADA (Anaby-Tavor *et al.*, 2020) e o PREDATOR (Abonizio; Junior, 2020), a abordagem proposta introduz distinções fundamentais. Enquanto o LAMBADA e o PREDATOR realizam o ajuste fino (*fine-tuning*) completo de seus respectivos modelos geradores, o método APS elimina esse passo, recorrendo ao pré-treinamento de LLMs e à configuração em *few-shot* para gerar exemplos adequados. Outra diferença fundamental ocorre em relação à geração de exemplos: o PREDATOR gera novos exemplos a partir de uma amostragem de exemplos-semente, enquanto o LAMBADA condiciona sua geração pela classe prefixada ao texto (cuja semântica é aprendida por ajuste). O método APS adota uma abordagem de geração baseada na instrução de LLMs sem necessidade de ajustar a instrução ao domínio. Neste processo de geração de paráfrases aplicamos a LLM apoiando-se no mecanismo nativo de preservação de classe (*label preservation*) (Bayer; Kaufhold; Reuter, 2022; Pellicer; Ferreira; Costa, 2023), visto que a LLM é instruída a manter a integridade semântica e classe do exemplo-semente. A abordagem sequencial do método APS se assemelha ao processo do LAMBADA, gerando os exemplos necessários e filtrando-os sequencialmente, iterando apenas uma vez nesse processo. Essa abordagem sequencial se distancia do PREDATOR, em que o processo de geração e filtragem ocorre de forma iterativa até atingir a quantidade necessária de exemplos. Ademais, ao contrário de seus predecessores, que operam com filtros baseados na probabilidade do exemplo pertencer à classe de interesse, o método proposto inclui dois métodos de seleção, baseados em diferentes abordagens de seleção de instâncias. A formalização deste fluxo é apresentada no Algoritmo 1.

Como indicado no Algoritmo 1, a abstração do método baseia-se em duas funções principais: a função geradora G e a função seletora S . A fim de detalhar a mecânica

Algoritmo 1 Fluxo de Operação do Método APS

Entrada: Base de treinamento original $D = \{D_{maj}, D_{min}\}$, Método Gerador G , Método de Seleção S , Proporção Alvo α

Saída: Conjunto final de exemplos sintéticos selecionados D_{sel}

- 1: Calcular o número de paráfrases k por instância minoritária com base em α
- 2: $D_{syn} \leftarrow G(D_{min}, k)$ {Geração massiva de paráfrases candidatas}
- 3: $D_{sel} \leftarrow S(D, D_{syn}, \alpha)$ {Aplicação da heurística de filtragem por Entropia ou Protótipo}
- 4: **retorne** D_{sel}

interna de cada etapa, o Algoritmo 2 expande o comportamento da função $G(D_{min}, k)$ ao operar sobre o conjunto minoritário, enquanto os Algoritmos 3 e 4 materializam as duas variantes metodológicas propostas para a função de filtragem $S(D, D_{syn}, \alpha)$. Os Algoritmos 2, 3 e 4 serão apresentados e explicados nas seções seguintes.

5.2 Paráfrase por meio de LLMs

O primeiro componente do método proposto consiste na geração de paráfrases por meio de uma LLM. Essa etapa apoia-se no método apresentado na Seção 4.2.4, mantendo-se o mesmo modelo, implementação e parâmetros de geração descritos anteriormente. De forma análoga, assume-se nesta etapa que todas as instâncias contidas em D_{min} possuem igual importância, portanto, contribuindo igualmente para o número de instâncias aumentadas.

Para cada exemplo original $e \in D_{min}$, a LLM recebe a instrução do sistema para gerar paráfrases, seguida da sentença e a ser parafraseada. A saída textual do modelo é processada e armazenada em um conjunto P_e . O *prompt* utilizado foi o mesmo especificado no Quadro 1 do Capítulo 4.

Como descrito no Capítulo 4 e no Quadro 1, o processo de geração é limitado a cinco paráfrases por inferência, para se adequar ao limite de contexto do modelo; portanto, o processo é executado iterativamente até que o conjunto atinja a cardinalidade ideal de $|P_e| = j \times k$. Nos experimentos presentes nessa dissertação, o valor escolhido para j é igual a 3. O parâmetro k determina o número necessário de instâncias sintéticas para cada exemplo da classe minoritária e, diferentemente do método apresentado no Capítulo 4, é calculado a partir da proporção alvo α . Quando a Razão de Desbalanceamento (RD) não é um valor inteiro ($RD \notin \mathbb{Z}$), a equação de igualdade estrita não pode ser satisfeita; escolhe-se, portanto, o maior $k \in \mathbb{Z}$ que satisfaça o limite superior estabelecido:

$$k \times |D_{min}| \leq \alpha \times |D_{maj}| - |D_{min}| \quad (5.1)$$

Calculamos então o parâmetro k da seguinte maneira:

$$k = \left\lfloor \frac{\alpha \times |D_{maj}| - |D_{min}|}{|D_{min}|} \right\rfloor \quad (5.2)$$

O intuito de gerar $j \times k$ exemplos candidatos para cada instância original é expandir a diversidade léxica disponível, fornecendo exemplos suficientes para que os seletores heurísticos operem de forma eficiente na etapa posterior, mesmo em configurações de balanceamento completo ($\alpha = 1,0$), em que o conjunto de paráfrases selecionadas compreenderia todas as paráfrases geradas se fossem gerados apenas $k \times e$ exemplos. A mecânica detalhada deste processo iterativo de geração é formalizada no Algoritmo 2.

Algoritmo 2 Função $G(D_{min}, k)$: Processo de Geração de Paráfrases

Entrada: Subconjunto minoritário D_{min} , Modelo Gerador LLM_G , Instrução do modelo ι , Parâmetro de expansão k

Saída: Paráfrases candidatas D_{syn}

```

1:  $D_{syn} \leftarrow \{\}$ 
2: para cada exemplo  $e \in D_{min}$  faça
3:    $P_e \leftarrow \{\}$ 
4:   enquanto  $|P_e| < j \times k$  faça
5:      $S_{gen} \leftarrow LLM_G(\iota, e)$  {Geração iterativa restrita pelo limite de contexto}
6:      $P_e \leftarrow P_e \cup S_{gen}$ 
7:      $P_e \leftarrow \text{RemoverDuplicata}(P_e)$ 
8:   fim enquanto
9:    $D_{syn} \leftarrow D_{syn} \cup P_e$ 
10: fim para
11: retorne  $D_{syn}$ 

```

5.3 Seleção de Paráfrases

A seleção de instâncias surge como um componente de filtragem pós-geração, cujo objetivo é mensurar a utilidade de cada exemplo sintético e reter aqueles que podem contribuir para melhorar o desempenho do classificador final.

A lógica de selecionar subconjuntos de dados com base em sua relevância é um conceito compartilhado por diferentes áreas do aprendizado de máquina (Snell; Swersky; Zemel, 2017; Lewis; Gale, 1994; Fonseca *et al.*, 2025). No Aprendizado Ativo (*Active Learning*) (Settles, 2009), a seleção é aplicada sobre um conjunto de dados não rotulados

para identificar quais instâncias reais são mais informativas e devem ser priorizadas para a anotação humana. No Aprendizado Baseado em Poucos Exemplos (*Few-Shot Learning*), a seleção ocorre na entrada do processo, escolhendo os melhores exemplos de suporte que comporão o contexto (*prompt*) para guiar a inferência de um modelo de linguagem. De maneira similar, a área de seleção de instâncias (*Instance Selection*), em que o objetivo é reduzir o volume de dados mantendo a representatividade da base em cenários de abundância de dados, tornando o processo de treinamento mais eficiente (Fonseca *et al.*, 2025).

No contexto do aumento de dados para classes desbalanceadas, a seleção de instâncias é usada para filtrar os exemplos mais informativos para o treinamento do modelo classificador. Para operacionalizar essa triagem, o processo baseia-se na aplicação de uma função ou métrica de importância, denotada por $M(x)$, que atribui uma pontuação escalar a cada instância candidata $x \in D_{syn}$. O subconjunto selecionado (D_{sel}) é obtido isolando as instâncias que satisfazem um critério de corte γ :

$$D_{sel} = \{x \in D_{syn} \mid M(x) \geq \gamma\} \quad (5.3)$$

O limiar γ é calculado a partir da cardinalidade-alvo, filtrando um conjunto de exemplos sintéticos de modo que $|D_{sel}| = \lfloor \alpha \times |D_{maj}| - |D_{min}| \rfloor$. De outro modo, pode-se definir o conjunto D_{sel} como o resultado de uma ordenação decrescente seguida de uma seleção dos $\lfloor \alpha \times |D_{maj}| - |D_{min}| \rfloor$ primeiros elementos do conjunto. Note que γ é calculado implicitamente por meio do valor do último elemento selecionado.

A definição matemática e o comportamento da métrica $M(x)$ determinam a natureza da filtragem, avaliando critérios distintos como o alinhamento de domínio no espaço latente ou a incerteza do classificador na fronteira de decisão. Esta dissertação explora dois métodos fundamentais de seleção, detalhados nas subseções a seguir.

5.3.1 Vetor Protótipo

A intuição subjacente a este método de seleção baseia-se na preservação da fidelidade semântica no espaço latente. Assume-se que instâncias sintéticas mapeadas próximas ao centro de massa da classe original retêm as propriedades discriminativas da categoria, ao passo que instâncias excessivamente distantes apresentam alta probabilidade de degradação semântica ou de troca inadvertida de classe (*label flipping*) (Natarajan *et al.*, 2013).

Com o intuito de filtrar as paráfrases semanticamente mais coerentes com a classe minoritária, este método fundamenta-se no conceito de vetor protótipo, extraído da literatura de clusterização e redes prototípicas (Snell; Swersky; Zemel, 2017), no qual o protótipo representa o centroide de um *cluster*. Sob esta perspectiva, o subconjunto minoritário original D_{min} é tratado como um *cluster*, do qual se extrai a representação central de referência. A Figura 5 ilustra a intuição do método no espaço de *embeddings*, os exemplos em vermelho são os minoritários enquanto os azuis são majoritários. O seletor prioriza exemplos próximos ao vetor protótipo (representado pelo ponto preto) que representa a classe minoritária como um todo.

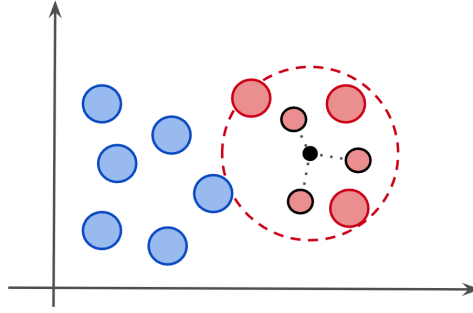


Figura 5: Método de seleção por similaridade ao vetor protótipo.

Para operacionalizar este seletor, os exemplos textuais em D_{min} são primeiramente projetados em um espaço de representação contínua (função E da Seção 2.1) por meio de um codificador denso baseado em *Transformers*. O modelo XLM-R foi escolhido para essa etapa, calculando a média dos *embeddings* na última camada. O vetor protótipo c_y é calculado como a média aritmética de todos os vetores de *embeddings* pertencentes ao conjunto minoritário:

$$c_y = \frac{1}{|D_{min}|} \sum_{x_i \in D_{min}} E(x_i) \quad (5.4)$$

Uma vez estabelecido o centroide c_y , cada paráfrase sintética candidata $x_j \in D_{syn}$ é igualmente mapeada no espaço latente por meio de $E(x_j)$. A função de importância $M(x_j)$ é definida pela similaridade de cosseno (*cosine similarity*) entre o vetor da paráfrase e o protótipo da classe, fornecendo uma métrica normalizada:

$$M(x_j) = \text{Sim}_{\cos}(E(x_j), c_y) = \frac{E(x_j) \cdot c_y}{\|E(x_j)\| \|c_y\|} \quad (5.5)$$

Por fim, as instâncias em D_{syn} são ordenadas decrescentemente pela pontuação de

similaridade, e os exemplos com as maiores pontuações são selecionados até atingir o volume estipulado por $|D_{sel}|$. O fluxo algorítmico desta abordagem está formalizado no Algoritmo 3.

Algoritmo 3 Função S : Seleção via Vetor Protótipo

Entrada: Base original $D = \{D_{maj}, D_{min}\}$, Paráfrases geradas D_{syn} , Proporção alvo α , Codificador E

Saída: Subconjunto de paráfrases filtradas $D_{sel} \subseteq D_{syn}$

```

1:  $N_{alvo} \leftarrow \lfloor \alpha \times |D_{maj}| - |D_{min}| \rfloor$  {Cálculo da quota de balanceamento}
2:  $V_{orig} \leftarrow \{\}$ 
3: para cada exemplo  $x_i \in D_{min}$  faça
4:    $V_{orig} \leftarrow V_{orig} \cup \{E(x_i)\}$ 
5: fim para
6:  $c_y \leftarrow \text{Média}(V_{orig})$  {Cálculo do centróide geométrico da classe}
7:  $Scores \leftarrow \{\}$ 
8: para cada paráfrase candidata  $x_j \in D_{syn}$  faça
9:    $s_j \leftarrow \text{Sim}_{\cos}(E(x_j), c_y)$ 
10:   $Scores \leftarrow Scores \cup \{(x_j, s_j)\}$ 
11: fim para
12:  $Scores\_Ordenado \leftarrow \text{OrdenarDecrescente}(Scores, \text{chave} = s_j)$ 
13:  $D_{sel} \leftarrow \text{SelecionarTopN}(Scores\_Ordenado, N_{alvo})$ 
14: retorne  $D_{sel}$ 

```

5.3.2 Entropia de Shannon

Este método fundamenta-se na amostragem por incerteza (*uncertainty sampling*) (Settles, 2009), transpondo o conceito de maximização da informatividade para o cenário de dados sintéticos. O princípio básico desta abordagem consiste em priorizar instâncias geradas que se posicionem em regiões de alta ambiguidade para um classificador auxiliar, denominado modelo de sondagem (*probe*). A Figura 6 ilustra a seleção do método no espaço de *embeddings*, os exemplos em vermelho são os minoritários enquanto os azuis são majoritários. O seletor prioriza exemplos próximos à fronteira de decisão, representada pela linha pontilhada.

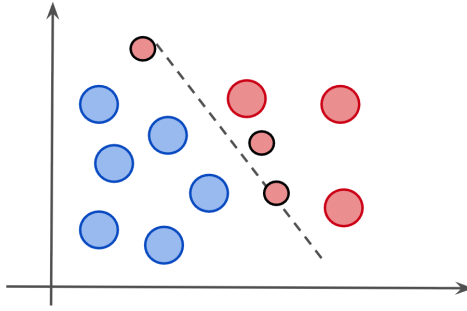


Figura 6: Método de seleção por entropia.

A incerteza preditiva é classicamente quantificada a partir dos vetores de probabilidade condicional gerados pelo modelo de sondagem. Adota-se para métrica de seleção $M(x)$ a Entropia de Shannon (H) como métrica formal de incerteza. A entropia avalia a distribuição de probabilidade de forma conjunta sobre todo o espaço discreto de classes Y :

$$H(x) = - \sum_{y_i \in Y} P(y_i | x) \log_2 P(y_i | x) \quad (5.6)$$

O processo operacional de seleção baseada em entropia é estruturado em três etapas consecutivas: (i) um classificador preditivo C é ajustado sobre o conjunto de dados de treinamento original D , gerando uma instância ajustada do classificador C_t ; (ii) as paráfrases sintéticas contidas em D_{syn} são submetidas ao processo de inferência sob o modelo C_t . A partir da distribuição de probabilidades retornada após a função *softmax*, calcula-se o valor escalar da entropia $H(x)$ para cada exemplo candidato; (iii) O conjunto de paráfrases candidatas é ordenado de forma decrescente com base nos valores de entropia obtidos, selecionando os exemplos de maior valor até atingir o número de exemplos necessário.

Para essa dissertação, o classificador C escolhido foi a regressão logística, devido à sua simplicidade e ao seu baixo custo computacional (Cunha; Rocha; Gonçalves, 2025). Para transformar os textos em um formato compatível para o classificador, foi usada a mesma estratégia de extração de atributos do Vetor Protótipo, usando o modelo XLM-R como função E . A intuição matemática deste critério reside no fato de que amostras que apresentam valores elevados de entropia aproximam-se de uma distribuição probabilística uniforme, caracterizando os cenários de máxima indefinição para o modelo. Ao priorizar estes exemplos na composição final de D_{sel} , o esforço de aumento de dados é concentrado estrategicamente ao longo da fronteira de decisão, otimizando sua capacidade de generalização e separabilidade (Lewis; Gale, 1994). A formalização matemática deste fluxo operacional é descrita no Algoritmo 4.

Algoritmo 4 Função S : Seleção via Entropia de Shannon

Entrada: Base original $D = \{D_{maj}, D_{min}\}$, Paráfrases geradas D_{syn} , Proporção alvo α , Classificador C

Saída: Subconjunto de paráfrases filtradas $D_{sel} \subseteq D_{syn}$

- 1: $N_{alvo} \leftarrow \lfloor \alpha \times |D_{maj}| - |D_{min}| \rfloor$ {Cálculo da quota de balanceamento}
 - 2: $C_t \leftarrow \text{TreinarClassificador}(C, D)$ {Ajuste do modelo *probe* na base original}
 - 3: $Scores \leftarrow \{\}$
 - 4: **para** cada paráfrase candidata $x_j \in D_{syn}$ **faça**
 - 5: Mapear a distribuição condicional $P(Y | x_j)$ por meio de C_t
 - 6: $h_j \leftarrow - \sum_{y_i \in Y} P(y_i | x_j) \log_2 P(y_i | x_j)$
 - 7: $Scores \leftarrow Scores \cup \{(x_j, h_j)\}$
 - 8: **fim para**
 - 9: $Scores_Ordenado \leftarrow \text{OrdenarDecrescente}(Scores, \text{chave} = h_j)$
 - 10: $D_{sel} \leftarrow \text{SelecionarTopN}(Scores_Ordenado, N_{alvo})$
 - 11: **retorne** D_{sel}
-

6 Avaliação do Método de Aumento por Paráfrases Seleccionadas

Este capítulo apresenta a avaliação do método proposto, comparando-o com outras importantes estratégias de balanceamento de classes. A Seção 6.1 detalha a metodologia experimental. As Seções 6.2 a 6.4 discutem os resultados dos experimentos. Por fim, a Seção 6.5 sumariza os principais achados do presente estudo.

6.1 Metodologia Experimental

Para avaliar o método proposto¹, foi expandida a metodologia aplicada no Capítulo 4. Foram mantidas as mesmas bases, métodos de balanceamento e estratégias de classificação. O estudo foi expandido para comparar o comportamento dos métodos em diferentes cenários de balanceamento ($\alpha = 0,5$ e $\alpha = 1,0$). As subseções a seguir detalham os aspectos expandidos da metodologia do Capítulo 4.

6.1.1 Baselines de Seleção de Paráfrase

Além de incluir os métodos avaliados anteriormente no Capítulo 4, e o método proposto. Também foram incluídas duas abordagens de seleção de paráfrases, consideradas duas formas básicas de seleção: PRIMEIROS k e SELEÇÃO ALEATÓRIA. O primeiro *baseline*, Primeiros k , é o método de paráfrases introduzido no estudo preliminar, detalhado no Capítulo 4, no qual são seleccionadas as primeiras k paráfrases de saída do LLM para cada exemplo original e oriundo do conjunto de treinamento. O segundo *baseline* seleciona aleatoriamente k paráfrases geradas para cada exemplo a partir do conjunto P_e de paráfrases geradas para e , com $|P_e| = k$. Para minimizar o risco da natureza aleatória do seletor conduzir a conclusões espúrias, esse experimento foi replicado cinco vezes, com sementes aleatórias para a seleção dos k exemplos. São reportados a média entre as execuções e o desvio padrão agregado.

¹Implementação disponível em: <https://github.com/Allrthur/APS>

6.1.2 Análise Quantitativa de Exemplos Gerados

A análise de exemplos gerados foi expandida a partir da metodologia do Capítulo 4 para se basear em duas métricas de geração de texto. Ambas as métricas fazem uso de um par candidato-referência (neste caso, texto original-exemplo gerado). Nesta avaliação, o exemplo gerado é medido em relação ao texto original em dois eixos: diversidade lexical e similaridade semântica. Para o primeiro eixo, foi escolhida a distância de Levenshtein normalizada (NED, do inglês, *Normalized Levenshtein Distance* ou *Normalized Edit Distance*) (Levenshtein, 1966). Essa métrica calcula a distância de edição entre o par candidato-referência, medindo quantas operações de remoção, adição e substituição são necessárias para transformar o texto-candidato no texto-referência. Essa distância é normalizada pelo tamanho do maior exemplo do par candidato-referência conforme

$$NED(C, R) = \frac{\text{Levenshtein}(C, R)}{\max(|C|, |R|)} \quad (6.1)$$

em que as sequências $C = c_1, c_2, \dots, c_n$ e $R = r_1, r_2, \dots, r_m$ representam texto-candidato e o texto-referência como uma sequência de caracteres, respectivamente. Uma forma intuitiva de entender distância de Levenshtein normalizada é como a proporção de remoções, adições e substituições para transformar o texto candidato na referência. Para o segundo eixo, foi escolhida a métrica *BERTScore* (Zhang et al., 2020), na qual um modelo *Transformer* transforma cada sentença do par em uma sequência de *embeddings* associados a cada *token*. Para cada par *token* candidato e *token* referência, é aplicada a similaridade de cosseno (Zhang et al., 2020) entre os *embeddings*. A métrica calcula seu resultado em três submétricas: precisão, que considera o maior resultado de cada *token* candidato em relação ao *token* referência; revocação, que considera o maior resultado de cada *token* referência em relação ao candidato e F1, a média harmônica entre a precisão e a revocação. Foi adotado o F1 como saída principal. Para facilitar a compreensão dos valores, aplicou-se o ajuste de escala em relação a um *baseline* (*rescale with baseline*), calculado com base nos resultados do *BERTScore* em uma base de pares candidato-referência. Esse ajuste requer que o modelo *Transformer* utilizado já tenha esse resultado previamente calculado. Portanto, para possibilitar o uso dessa configuração, utilizou-se o modelo padrão da métrica em inglês (*roberta-large* (Liu et al., 2019)).

6.2 Análise de Desempenho Preditivo

As Tabelas 7 a 14 detalham os resultados de *Macro-F1* (média $\pm$ desvio padrão considerando os 10 *folds*) do processo de validação cruzada, nas nove bases testadas, com exceção da seleção aleatória, que, por ser o resultado de cinco repetições, reporta a média de médias e desvio padrão agregado. Números em negrito indicam o melhor desempenho médio da base. Já números sublinhados indicam significância estatística em relação ao cenário sem balanceamento por meio do teste de *Wilcoxon Rank-Sum* (Wilcoxon, 1945), com significância estatística de $p < 0,05$. Também é reportada a posição média no *ranking* de resultados de cada estratégia (i.e., a média da posição da estratégia em cada base avaliada, em que menores valores indicam melhores posições). Seguindo a metodologia do Capítulo 4, essa análise foi feita com duas estratégias de classificação, as subseções seguintes detalham os resultados de desempenho preditivo das duas estratégias de classificação, de forma individual (LR e XLM-R).

6.2.1 Desempenho com LR

As Tabelas 7 e 8 comparam os dois métodos de seleção propostos aos métodos básicos de seleção de paráfrases. Os resultados do método APS, instanciado com seleção por entropia, superam os resultados de ambos os *baselines* e do APS instanciado com seleção pelo vetor protótipo, atingindo o melhor desempenho médio em seis das nove bases com $\alpha = 0,5$ e em todas as bases em $\alpha = 1,0$.

Tabela 7: Desempenho preditivo entre seletores de paráfrase com $\alpha = 0,5$ (LR)

Base	Primeiros K	Aleatório (Agr)	APS (Protótipo)	APS (Entropia)
CI/SST-2	0,6037 $\pm$ 0,0124	0,6034 $\pm$ 0,0127	0,5875 $\pm$ 0,0102	0,6346 $\pm$ 0,0195
CI/SemEval 2017	0,6290 $\pm$ 0,0368	0,6264 $\pm$ 0,0393	0,6431 $\pm$ 0,0449	0,6685 $\pm$ 0,0584
CI/IMDB	0,6050 $\pm$ 0,0261	0,6037 $\pm$ 0,0245	0,6308 $\pm$ 0,0171	0,6214 $\pm$ 0,0556
CI/Rotten Tomatoes	0,6045 $\pm$ 0,0357	0,6025 $\pm$ 0,0377	0,6110 $\pm$ 0,0448	0,6223 $\pm$ 0,0425
SemEval 2015	0,6253 $\pm$ 0,1625	0,6173 $\pm$ 0,1590	0,5916 $\pm$ 0,1611	0,6106 $\pm$ 0,1620
SemEval 2024	0,6857 $\pm$ 0,0230	0,6836 $\pm$ 0,0259	0,6800 $\pm$ 0,0233	0,6912 $\pm$ 0,0279
SemEval 2025	0,6200 $\pm$ 0,0505	0,6228 $\pm$ 0,0457	0,6248 $\pm$ 0,0482	0,6231 $\pm$ 0,0384
Twitter Topics (0 vs All)	0,6411 $\pm$ 0,0605	0,6494 $\pm$ 0,0628	0,6700 $\pm$ 0,0531	0,6775 $\pm$ 0,0452
Twitter Topics (1 vs All)	0,7628 $\pm$ 0,0358	0,7562 $\pm$ 0,0402	0,7750 $\pm$ 0,0370	0,7955 $\pm$ 0,0328
Posto Médio	2,78	3,33	2,44	1,44

As Tabelas 9 e 10 comparam os métodos de balanceamento combinados com o modelo de regressão logística quando α é igual a 0,5 e 1,0, respectivamente. Nesta tabela, o método proposto é representado pela versão com seleção por entropia pois foi a que apresentou o melhor resultado na avaliação anterior. Os métodos SMOTE (1,22) e APS

Tabela 8: Desempenho preditivo entre seletores de paráfrase com $\alpha = 1,0$ (LR)

Base	Primeiros K	Aleatório (Agr)	APS (Protótipo)	APS (Entropia)
CI/SST-2	$0,5744 \pm 0,0071$	$0,5753 \pm 0,0061$	$0,5708 \pm 0,0086$	$0,6384 \pm 0,0125$
CI/SemEval 2017	$0,6037 \pm 0,0228$	$0,6042 \pm 0,0265$	$0,6106 \pm 0,0341$	$0,6766 \pm 0,0578$
CI/IMDB	$0,6114 \pm 0,0192$	$0,6105 \pm 0,0237$	$0,6261 \pm 0,0175$	$0,6481 \pm 0,0407$
CI/Rotten Tomatoes	$0,5995 \pm 0,0392$	$0,6047 \pm 0,0408$	$0,6048 \pm 0,0395$	$0,6329 \pm 0,0420$
SemEval 2015	$0,6539 \pm 0,1276$	$0,6500 \pm 0,1270$	$0,6437 \pm 0,1462$	$0,7022 \pm 0,1081$
SemEval 2024	$0,6751 \pm 0,0246$	$0,6733 \pm 0,0231$	$0,6746 \pm 0,0274$	$0,6932 \pm 0,0279$
SemEval 2025	$0,6218 \pm 0,0391$	$0,6214 \pm 0,0406$	$0,6062 \pm 0,0344$	$0,6596 \pm 0,0479$
Twitter Topics (0 vs All)	$0,6335 \pm 0,0491$	$0,6355 \pm 0,0478$	$0,6585 \pm 0,0421$	$0,6827 \pm 0,0491$
Twitter Topics (1 vs All)	$0,7320 \pm 0,0331$	$0,7360 \pm 0,0327$	$0,7489 \pm 0,0370$	$0,7937 \pm 0,0317$
Posto Médio	3,11	3,11	2,78	1,00

Tabela 9: Desempenho preditivo entre métodos de TDA com $\alpha = 0,5$ (LR)

Base	Sem Balanceamento	SMOTE	EDA	PREDATOR	APS (Entropia)
CI/SST-2	$0,5257 \pm 0,0133$	$0,6141 \pm 0,0080$	$0,5924 \pm 0,0131$	$0,5650 \pm 0,0104$	$0,6346 \pm 0,0195$
CI/SemEval 2017	$0,5170 \pm 0,0346$	$0,6362 \pm 0,0505$	$0,5742 \pm 0,0532$	$0,5733 \pm 0,0550$	$0,6685 \pm 0,0584$
CI/IMDB	$0,5099 \pm 0,0091$	$0,6759 \pm 0,0172$	$0,5499 \pm 0,0434$	$0,6176 \pm 0,0425$	$0,6214 \pm 0,0556$
CI/Rotten Tomatoes	$0,5110 \pm 0,0251$	$0,6955 \pm 0,0290$	$0,5936 \pm 0,0438$	$0,6113 \pm 0,0494$	$0,6223 \pm 0,0425$
SemEval 2015	$0,5195 \pm 0,0982$	$0,6989 \pm 0,1388$	$0,6428 \pm 0,1514$	$0,6711 \pm 0,1501$	$0,6106 \pm 0,1620$
SemEval 2024	$0,6658 \pm 0,0250$	$0,7128 \pm 0,0272$	$0,6792 \pm 0,0213$	$0,7044 \pm 0,0251$	$0,6912 \pm 0,0279$
SemEval 2025	$0,5241 \pm 0,0228$	$0,6639 \pm 0,0429$	$0,5858 \pm 0,0517$	$0,6458 \pm 0,0411$	$0,6231 \pm 0,0384$
Twitter Topics (0 vs All)	$0,5008 \pm 0,0199$	$0,6860 \pm 0,0208$	$0,5394 \pm 0,0562$	$0,6666 \pm 0,0734$	$0,6775 \pm 0,0452$
Twitter Topics (1 vs All)	$0,7350 \pm 0,0486$	$0,8025 \pm 0,0158$	$0,7245 \pm 0,0490$	$0,7938 \pm 0,0413$	$0,7955 \pm 0,0328$
Ranking Médio	4,89	1,22	3,78	2,89	2,22

Tabela 10: Desempenho preditivo entre métodos de TDA com $\alpha = 1,0$ (LR)

Base	Sem Balanceamento	SMOTE	EDA	PREDATOR	APS (Entropia)
CI/SST-2	$0,5257 \pm 0,0133$	$0,5773 \pm 0,0071$	$0,5656 \pm 0,0095$	$0,5412 \pm 0,0074$	$0,6384 \pm 0,0125$
CI/SemEval 2017	$0,5170 \pm 0,0346$	$0,6182 \pm 0,0274$	$0,5757 \pm 0,0559$	$0,5722 \pm 0,0623$	$0,6766 \pm 0,0578$
CI/IMDB	$0,5099 \pm 0,0091$	$0,6165 \pm 0,0151$	$0,5521 \pm 0,0473$	$0,6146 \pm 0,0371$	$0,6481 \pm 0,0407$
CI/Rotten Tomatoes	$0,5110 \pm 0,0251$	$0,6544 \pm 0,0225$	$0,5966 \pm 0,0369$	$0,6131 \pm 0,0353$	$0,6329 \pm 0,0420$
SemEval 2015	$0,5195 \pm 0,0982$	$0,7731 \pm 0,1185$	$0,6433 \pm 0,1634$	$0,6983 \pm 0,1548$	$0,7022 \pm 0,1081$
SemEval 2024	$0,6658 \pm 0,0250$	$0,7045 \pm 0,0174$	$0,6752 \pm 0,0259$	$0,7084 \pm 0,0224$	$0,6932 \pm 0,0279$
SemEval 2025	$0,5241 \pm 0,0228$	$0,6588 \pm 0,0274$	$0,5970 \pm 0,0646$	$0,6693 \pm 0,0450$	$0,6596 \pm 0,0479$
Twitter Topics (0 vs All)	$0,5008 \pm 0,0199$	$0,6522 \pm 0,0193$	$0,5478 \pm 0,0670$	$0,6641 \pm 0,0810$	$0,6827 \pm 0,0491$
Twitter Topics (1 vs All)	$0,7350 \pm 0,0486$	$0,7608 \pm 0,0186$	$0,7210 \pm 0,0448$	$0,7850 \pm 0,0432$	$0,7937 \pm 0,0317$
Ranking Médio	4,89	2,11	3,89	2,56	1,56

(2,22) obtiveram os dois melhores valores de *ranking* médio: o SMOTE obteve melhor *ranking* médio quando $\alpha = 0,5$ enquanto, quando $\alpha = 1,0$, o método proposto lidera o *ranking* médio (1,56).

Para confirmar se as diferenças entre os resultados obtidos são significativas, foi realizado um teste estatístico de Friedman com $p \leq 0,05$, seguido do teste *post-hoc* de Nemenyi para cada comparativo presente nas Tabelas 7 a 10. Essa combinação foi escolhida por ser amplamente recomendada para a comparação simultânea de múltiplos algoritmos em diferentes bases de dados (Demšar, 2006). Na primeira análise, o teste de Friedman confirmou diferenças significativas entre as estratégias de seleção de paráfrase (Friedman = 10,2000 e $p = 1,6940 \times 10^{-2}$ para $\alpha = 0,5$ e Friedman = 16,6000 e $p = 8,5404 \times 10^{-4}$ para $\alpha = 1,0$). A Figura 7 demonstra os resultados par a par do teste *post-hoc* de Nemenyi em que células vermelhas indicam valores $p \geq 0,05$ e células em azul indicam valores $p < 0,05$. A seleção baseada em entropia do método proposto se destaca em relação aos outros métodos de seleção quando combinada à regressão logística, demonstrando uma diferença significativa de desempenho entre os demais métodos de seleção em $\alpha = 1,0$ e mostrando uma diferença significativa em relação ao método de seleção aleatória em $\alpha = 0,5$.

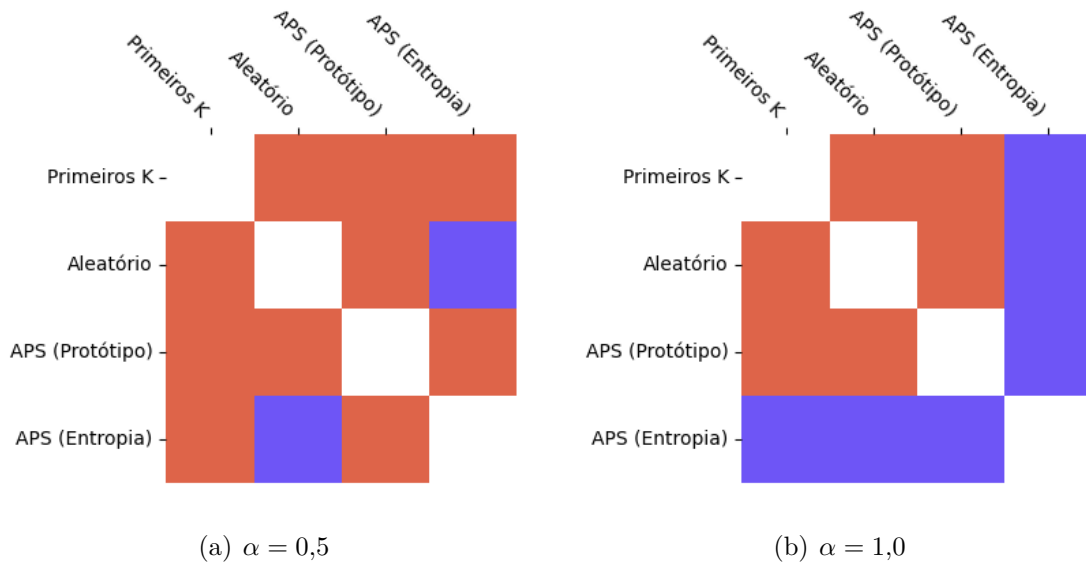


Figura 7: Análise *post-hoc* de Nemenyi entre métodos de seleção de paráfrase (LR). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.

Analisando o comparativo entre métodos de balanceamento, o teste de Friedman confirmou diferenças significativas (Friedman = 28,6222 e $p = 9,2376 \times 10^{-6}$ para $\alpha = 0,5$ e Friedman = 26,7556 e $p = 2,2274 \times 10^{-5}$ para $\alpha = 1,0$). A Figura 8 denota o valor p par a par entre os métodos de balanceamento. Observa-se que, com $\alpha = 0,5$, o SMOTE

apresenta diferenças significativas de desempenho em relação aos demais métodos, com exceção do método proposto. Já com $\alpha = 1,0$ o teste não encontra diferenças significativas entre o SMOTE e os demais métodos, com exceção do EDA.

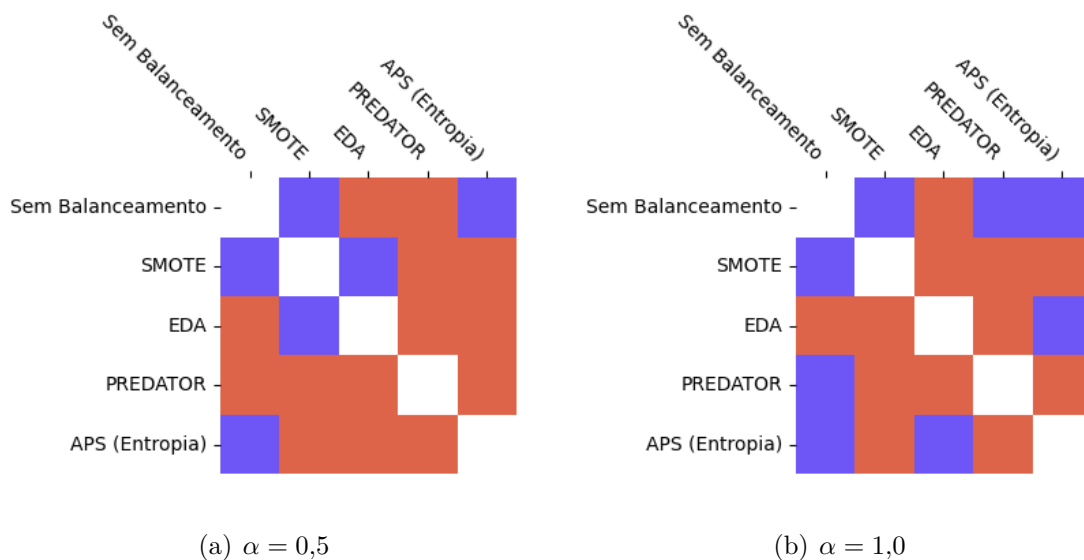


Figura 8: Análise *post-hoc* de Nemenyi entre métodos de balanceamento (LR). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.

Observa-se pela Tabela 9 e 10 que SMOTE e APS apresentam o desempenho preditivo médio mais alto em $\alpha = 0,5$ e $\alpha = 1,0$, respectivamente. O teste *post-hoc* encontra diferenças significativas entre SMOTE e APS em relação à base desbalanceada, independentemente de α . No entanto, esses resultados não encontram diferenças significativas entre os dois métodos.

6.2.2 Desempenho com XLM-R

Replicando a metodologia da subseção anterior, as Tabelas 11 e 12 comparam os métodos de seleção propostos com outras formas de seleção de paráfrase, considerando o modelo de classificação XLM-R. Similarmente aos resultados da regressão logística, a seleção por entropia demonstra melhor desempenho médio na maioria das bases (seis bases com $\alpha = 0,5$ e cinco bases com $\alpha = 1,0$). As Tabelas 13 e 14 comparam o método de paráfrase proposto (novamente, representado pela versão com seleção por entropia) com os outros métodos de balanceamento. O método proposto demonstra desempenho médio superior em cinco bases em $\alpha = 0,5$ e em sete bases em $\alpha = 1,0$.

De forma similar às análises da subseção anterior, aplicou-se o teste de Friedman

Tabela 11: Análise entre seletores de paráfrase com $\alpha = 0,5$ (XLM-R)

Base	Primeiros k	Aleatório (Agr)	APS (Protótipo)	APS (Entropia)
CI/SST-2	$0,7267 \pm 0,0252$	$0,7310 \pm 0,0273$	$0,7031 \pm 0,0321$	$0,7480 \pm 0,0301$
CI/SemEval 2017	$0,7045 \pm 0,0657$	$0,6892 \pm 0,0658$	$0,6899 \pm 0,0558$	$0,7133 \pm 0,0453$
CI/IMDB	$0,6999 \pm 0,0393$	$0,6864 \pm 0,0552$	$0,7135 \pm 0,0557$	$0,7245 \pm 0,0472$
CI/Rotten Tomatoes	$0,5750 \pm 0,0929$	$0,5602 \pm 0,0947$	$0,6315 \pm 0,1140$	$0,6186 \pm 0,1110$
SemEval 2015	$0,5722 \pm 0,1613$	$0,6589 \pm 0,1636$	$0,6517 \pm 0,1442$	$0,6347 \pm 0,1665$
SemEval 2024	$0,6865 \pm 0,0407$	$0,6737 \pm 0,0586$	$0,6903 \pm 0,0296$	$0,7054 \pm 0,0255$
SemEval 2025	$0,6125 \pm 0,0802$	$0,6482 \pm 0,0615$	$0,6417 \pm 0,0654$	$0,6363 \pm 0,1001$
Twitter Topics (0 vs All)	$0,6673 \pm 0,0911$	$0,6728 \pm 0,0603$	$0,6951 \pm 0,0670$	$0,7038 \pm 0,0711$
Twitter Topics (1 vs All)	$0,8016 \pm 0,0350$	$0,7966 \pm 0,0337$	$0,8041 \pm 0,0361$	$0,8166 \pm 0,0388$
Ranking Médio	3,22	3,00	2,22	1,56

Tabela 12: Análise entre seletores de paráfrase com $\alpha = 1,0$ (XLM-R)

Base	Primeiros k	Aleatório (Agr)	APS (Protótipo)	APS (Entropia)
CI/SST-2	$0,7313 \pm 0,0190$	$0,7148 \pm 0,0321$	$0,7125 \pm 0,0163$	$0,7501 \pm 0,0280$
CI/SemEval 2017	$0,6968 \pm 0,0620$	$0,6902 \pm 0,0609$	$0,6878 \pm 0,0670$	$0,6884 \pm 0,0964$
CI/IMDB	$0,6721 \pm 0,0450$	$0,6729 \pm 0,0966$	$0,6879 \pm 0,0436$	$0,7047 \pm 0,0538$
CI/Rotten Tomatoes	$0,5755 \pm 0,0930$	$0,6031 \pm 0,1161$	$0,6158 \pm 0,1070$	$0,5857 \pm 0,1240$
SemEval 2015	$0,6343 \pm 0,1571$	$0,6897 \pm 0,1562$	$0,7117 \pm 0,1111$	$0,6934 \pm 0,1520$
SemEval 2024	$0,6600 \pm 0,0315$	$0,6591 \pm 0,0398$	$0,6338 \pm 0,0447$	$0,6839 \pm 0,0334$
SemEval 2025	$0,6146 \pm 0,0437$	$0,6321 \pm 0,0499$	$0,6175 \pm 0,0519$	$0,6685 \pm 0,0383$
Twitter Topics (0 vs All)	$0,6651 \pm 0,0700$	$0,6445 \pm 0,0926$	$0,6845 \pm 0,0726$	$0,7078 \pm 0,0811$
Twitter Topics (1 vs All)	$0,8049 \pm 0,0497$	$0,8060 \pm 0,0348$	$0,8303 \pm 0,0415$	$0,8269 \pm 0,0213$
Ranking Médio	3,11	2,78	2,44	1,67

Tabela 13: Desempenho preditivo entre métodos de TDA com $\alpha = 0,5$ (XLM-R)

Base	Sem Balanceamento	EDA	PREDATOR	APS (Entropia)
CI/SST-2	$0,7013 \pm 0,0476$	$0,7402 \pm 0,0251$	$0,6777 \pm 0,0196$	$0,7480 \pm 0,0301$
CI/SemEval 2017	$0,5865 \pm 0,0961$	$0,6711 \pm 0,1023$	$0,6143 \pm 0,1020$	$0,7133 \pm 0,0453$
CI/IMDB	$0,6785 \pm 0,0880$	$0,6779 \pm 0,0726$	$0,6966 \pm 0,0352$	$0,7245 \pm 0,0472$
CI/Rotten Tomatoes	$0,6255 \pm 0,1155$	$0,6100 \pm 0,1175$	$0,6536 \pm 0,0696$	$0,6186 \pm 0,1110$
SemEval 2015	$0,4792 \pm 0,0582$	$0,6722 \pm 0,1755$	$0,7017 \pm 0,1423$	$0,6347 \pm 0,1665$
SemEval 2024	$0,7030 \pm 0,0264$	$0,6381 \pm 0,0709$	$0,7051 \pm 0,0197$	$0,7054 \pm 0,0255$
SemEval 2025	$0,6343 \pm 0,0948$	$0,6355 \pm 0,0994$	$0,6880 \pm 0,0327$	$0,6363 \pm 0,1001$
Twitter Topics (0 vs All)	$0,6087 \pm 0,1175$	$0,6787 \pm 0,0995$	$0,6493 \pm 0,0869$	$0,7038 \pm 0,0711$
Twitter Topics (1 vs All)	$0,8201 \pm 0,0156$	$0,7957 \pm 0,0473$	$0,8122 \pm 0,0336$	$0,8166 \pm 0,0388$
Ranking Médio	3,11	3,00	2,22	1,67

Tabela 14: Desempenho preditivo entre métodos de TDA com $\alpha = 1,0$ (XLM-R)

Base	Sem Balanceamento	EDA	PREDATOR	APS (Entropia)
CI/SST-2	$0,7013 \pm 0,0476$	$0,7384 \pm 0,0320$	$0,6703 \pm 0,0238$	$0,7501 \pm 0,0280$
CI/SemEval 2017	$0,5865 \pm 0,0961$	$0,6486 \pm 0,0685$	$0,6220 \pm 0,0594$	$0,6884 \pm 0,0964$
CI/IMDB	$0,6785 \pm 0,0880$	$0,6639 \pm 0,1040$	$0,6690 \pm 0,0656$	$0,7047 \pm 0,0538$
CI/Rotten Tomatoes	$0,6255 \pm 0,1155$	$0,5209 \pm 0,0599$	$0,6398 \pm 0,0891$	$0,5857 \pm 0,1240$
SemEval 2015	$0,4792 \pm 0,0582$	$0,6772 \pm 0,1641$	$0,6541 \pm 0,1620$	$0,6934 \pm 0,1520$
SemEval 2024	$0,7030 \pm 0,0264$	$0,6877 \pm 0,0180$	$0,7026 \pm 0,0254$	$0,6839 \pm 0,0334$
SemEval 2025	$0,6343 \pm 0,0948$	$0,5607 \pm 0,0854$	$0,6632 \pm 0,0314$	$0,6685 \pm 0,0383$
Twitter Topics (0 vs All)	$0,6087 \pm 0,1175$	$0,6095 \pm 0,1048$	$0,6697 \pm 0,0829$	$0,7078 \pm 0,0811$
Twitter Topics (1 vs All)	$0,8201 \pm 0,0156$	$0,8127 \pm 0,0140$	$0,8116 \pm 0,0349$	$0,8269 \pm 0,0213$
Ranking Médio	2,78	3,00	2,67	1,56

seguido pelo teste *post-hoc* de Nemenyi. Primeiramente, comparando os métodos de seleção de paráfrases, o teste de Friedman aponta diferença significativa apenas entre os métodos de seleção em $\alpha = 0,5$ (Friedman = 9,4000 e $p = 2,4419 \times 10^{-2}$ para $\alpha = 0,5$ e Friedman = 6,2000 e $p = 1,0228 \times 10^{-1}$ para $\alpha = 1,0$).

A Figura 9 denota os valores p par a par obtidos pelos testes *post-hoc* de Nemenyi em ambos os cenários de balanceamento. A figura ilustra que não foram encontradas diferenças significativas entre os métodos de seleção independentemente de α , exceto pela seleção por entropia do método proposto em relação ao método de Primeiros k .

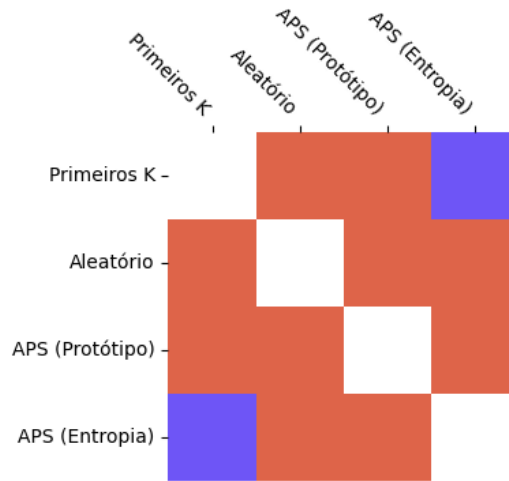
(a) $\alpha = 0,5$

Figura 9: Análise *post-hoc* de Nemenyi entre métodos de seleção de paráfrase (XLM-R). Células em azul indicam valor $p < 0,05$, enquanto células em vermelho denotam o caso contrário.

Uma segunda análise foi feita com base no comparativo entre métodos de balanceamento. Por meio do teste de Friedman, não foi notada uma diferença significativa entre

o desempenho dos métodos (Friedman = 7,5333 e $p = 5,6708 \times 10^{-2}$ para $\alpha = 0,5$ e Friedman = 6,7333 e $p = 8,0901 \times 10^{-2}$ para $\alpha = 1,0$). A Figura ??apresenta o valor p par a par entre os métodos obtidos.

6.3 Análise dos Exemplos Gerados

Para investigar os fatores que podem ter contribuído para os resultados observados, são apresentados exemplos qualitativos na Tabela 15, comparando as instâncias geradas aos respectivos exemplos originais. Foram escolhidos os exemplos das bases em que os métodos demonstraram melhor desempenho médio no estudo preliminar, e que foram superados pelo método proposto (CI/SST-2, CI/SemEval 2017 e Twitter Topics (0 vs All)). Para fundamentar o comparativo, foram calculadas as métricas de diversidade léxica (NED) e de similaridade semântica (BERTScore) para os pares analisados.

Tabela 15: Exemplos gerados pelas técnicas vencedoras do estudo anterior em comparação com a técnica vencedora Paráfrase selecionada por APS.

Base	Texto Original	TDA Vencedor	Texto Gerado
CI/SST-2	, you 're far better served by the source material .	EDA	you re far better served by the source textile
		APS	you're much better off sticking with the original content.
CI/SemEval 2017	Chris Evans just said exactly what I was thinking about Saturday Kitchen 'I don't care about this show anymore'	Paráfrase	Chris Evans just echoed my sentiments about Saturday Kitchen, I'm so over it now
		APS	I'm totally over Saturday Kitchen, can't even be bothered with it anymore
Twitter (0 vs All)	This book provides the perfect quick read Don t miss Truth and Grace Homeschool Academy s review of Ghosted at the Altar by {{USERNAME}} #book & enter the #free #giveaway for a \$25 Amazon gift card! #books #amreading {{URL}}	PREDATOR	great book so read up for you). I will be sharing the book with you as I read it on my blog {{URL}} and if you have questions like that, you can Giveaway alert! Win a \$25 Amazon gift card and check out Truth and Grace Homeschool Academy's review of Ghosted at the Altar by {{USERNAME}} #books #amreading {{URL}}
		APS	

No caso da base CI/SST-2, observa-se que o exemplo gerado pelo método APS (com seleção por entropia) introduz uma variedade léxica maior que as perturbações heurísticas do EDA. Embora o método APS tenha demonstrado uma diferença lexical maior (de $NED = 0,19$ para 0,63), a similaridade semântica se manteve semelhante (de 0,49 para 0,44, em BERTScore). Para a base CI/SemEval 2017, é comparada a saída da paráfrase sem mecanismo de seleção em relação ao método proposto, com seleção por entropia. Notavelmente, ambos os métodos geraram uma saída linguisticamente coerente e fluida.

No entanto, é possível notar que a paráfrase escolhida pelo método APS apresenta uma diferença lexical maior, pela troca da ordem das orações (de $NED = 0,50$ para $0,67$). Essa troca de ordem, contudo, resultou em uma perda de similaridade semântica em relação ao texto original ($BERTScore = 0,22$ contra $0,38$). Todavia, essa perda de similaridade semântica pode ser benéfica, ajudando o modelo a generalizar, desde que não afete a classificação do exemplo.

A saída do PREDATOR é frequentemente truncada, gerando frases incompletas. Em contrapartida, como consequência do ajuste de domínio, o PREDATOR gera exemplos fiéis ao estilo linguístico², ilustrados pela capitalização e pela manutenção de estruturas como $\{\{USERNAME\}\}$. No entanto, observa-se que a saída do método APS também apresenta a mesma aderência, replicando as estruturas $\{\{USERNAME\}\}$ e $\#books$. Contudo, devido à natureza da geração do PREDATOR, este método gera exemplos mais diversos lexicalmente em relação ao método APS ($NED = 0,75$ do PREDATOR contra $0,49$ do APS). No entanto, esse mesmo fator leva o método a gerar exemplos semanticamente desconexos ($BERTScore = -0,06$ no exemplo do PREDATOR, enquanto APS mantém similaridade semântica de $0,58$). Vale ressaltar, todavia, que essa mesma natureza do PREDATOR parece ter ajudado o classificador. Como supracitado, a perda de similaridade semântica pode ser benéfica, trazendo diversidade, desde que o exemplo gerado seja semanticamente representativo da classe de interesse.

A Tabela 16 complementa esta análise com os exemplos gerados pelos métodos vencedores nas demais bases, considerando apenas as bases em que os métodos com o melhor desempenho são aqueles cujas saídas são textos. No caso da base CI/IMDB, composta por resenhas longas, nota-se que o método proposto demonstra capacidade de manter o tom sarcástico e a polaridade da resenha original, reescrevendo metáforas de forma fluida, adaptando “goo that forms at the bottom of a trash can” para “fossilized piece of trash”, alcançando alta variância léxica ($NED = 0,62$) sem perder a âncora semântica do parágrafo ($BERTScore = 0,31$). Em contrapartida, o texto gerado pelo PREDATOR para a base SemEval 2025 ilustra as limitações arquiteturais do método. Apesar de obter métricas aceitáveis ($NED = 0,65$ e $BERTScore = 0,46$), a avaliação qualitativa revela o truncamento frequentemente observado (“for some reason”) e repetições (“humiliating humiliation” e “for some”).

Apesar da robustez geral, a análise do exemplo oriundo do Twitter Topics (1 vs All) revela uma limitação da técnica APS em textos puramente informativos. Embora o

²O PREDATOR não tem um exemplo original direto devido à sua arquitetura e, portanto, foi pareado com o exemplo de maior similaridade de cosseno de D_{min} .

método preserve com sucesso as restrições estruturais do domínio (como *USERNAME* e *#revenue*), a paráfrase gerada sofre uma simplificação excessiva, removendo orações do exemplo original. Essa supressão reflete-se diretamente nas métricas: o método proposto apresenta um elevado grau de variação lexical ($NED = 0,63$), mas desvia-se da semântica original ($BERTScore = 0,26$). De modo geral, a análise qualitativa indica que o método proposto se destaca pela fluidez linguística e pela capacidade de gerar variância estrutural. Contudo, enquanto a técnica evita truncamentos, perda grave de semântica e ruídos, ela pode tender à concisão em sentenças longas não-narrativas, suprimindo orações de menor importância.

Tabela 16: Exemplos gerados pelas técnicas vencedoras nas demais bases.

Base	Texto Original	TDA Vencedor	Texto Gerado
CI/IMDB	Go immediately and rent this movie. It will be be on a bottom shelf in your local video store and will be covered in dust. No one will have touched it in years. It may even be a \$.50 special! It's worth ten bucks, I swear! Buy it! There aren't very many films than can compare with this - the celluloid version of that goo that forms at the bottom of a trash can after a few years. Yes, I gave it a '1,' but it really deserves much lower. 1-10 scales were not designed with stuff like this in mind.	APS	Trust me, grab this movie ASAP, it's probably collecting dust on the bottom shelf at your local video store and will cost you a buck, it's worth ten times that, I swear! It's a cinematic gem that's in a league of its own, like a fossilized piece of trash that's been sitting in the can for years, yeah, I gave it a 1 but it's way worse than that, 1-10 just doesn't cut it for something this bad!
SemEval 2025	After the humiliation and my husband laughing uncontrollably at me, my nails remained perfectly intact.	PREDATOR	for some reason. After the humiliating humiliation and my husband laughing uncontrollably for some
Twitter (1 vs All)	Chinese on-demand delivery and retail platform {{USERNAME}} on Tuesday reported a 52% jump in first quarter #revenue from a year earlier, as the company tries to expand its #Network to the country s lower-tier cities and #Rural areas. {{URL}}	APS	the future is bright for {{USERNAME}}, with a 52% jump in first quarter #revenue that's got everyone talking about its potential

6.4 Análise de Custo

Com o intuito de investigar o custo computacional dos métodos, foi medido o tempo de processamento necessário para gerar os exemplos sintéticos. Para deixar mais evidente o impacto dos métodos no custo, a análise foca no cenário $\alpha = 1,0$. A medição de tempo dos métodos foi realizada por meio de bibliotecas padrão do *Python*, à exceção da abordagem baseada em paráfrases via LLM, cujo tempo de execução foi monitorado através da plataforma *Weights & Biases* (Biewald, 2020), que registra apenas segundos inteiros, sem frações decimais. Vale ressaltar que todos os experimentos foram executados na

Tabela 17: Tempo total para a geração de exemplos em segundos.

Base	SMOTE	EDA	PREDATOR	Paráfrase
CI/SST-2	1,79	4,24	2.156,37	48.884,00
CI/SemEval 2017	0,51	0,97	654,15	10.150,00
CI/IMDB	0,72	23,54	14.529,00	32.622,00
CI/Rotten Tomatoes	0,27	0,67	324,26	17.608,00
SemEval2015	0,34	1,18	683,62	668,00
SemEval2024	0,43	0,84	639,15	11.614,00
SemEval2025	0,26	0,38	480,69	5.168,00
Twitter (0 vs All)	0,32	1,39	4.726,22	25.079,00
Twitter (1 vs All)	0,31	1,65	2.568,87	25.214,00

mesma máquina de configuração (Intel Core Ultra 9 285K, 130 GB de memória RAM e GPU RTX 5090 com 32GB de VRAM) para que os tempos fossem diretamente comparáveis. A Tabela 17 ilustra o tempo decorrido para o processo de geração dos exemplos em segundos. Ao analisar estes resultados, percebe-se que os métodos PREDATOR e APS, por se valerem de modelos de linguagem generativos, acarretam um custo computacional muito superior ao dos métodos heurísticos (SMOTE e EDA). Entre os métodos de balanceamento generativos, APS apresenta um custo computacional em ordens de magnitude maiores do que os demais métodos, devido ao uso de uma LLM local de 8 bilhões de parâmetros. Dando contexto a esse número, o modelo gerador utilizado no PREDATOR (DistilGPT2) possui 82 milhões de parâmetros.

Outro determinante do tempo total de geração de exemplos pelo método APS é o número de instâncias necessárias para equilibrar cada base. Na base CI/SST-2, por exemplo, são necessários 33.144 exemplos sintéticos para atingir o balanceamento ($\alpha = 1,0$), resultando em um tempo expressivo de inferência. Em contrapartida, bases com poucos exemplos, como a SemEval 2015 (que demanda apenas 227 instâncias geradas), registram tempos totais menores, tornando a execução do APS competitiva em termos de tempo mesmo frente a métodos estruturalmente mais leves.

As Figuras 10 e 11 exibem a relação entre o ganho no desempenho preditivo médio dos métodos avaliados em relação ao cenário sem balanceamento e o tempo de geração, em segundos, na escala logarítmica. O modelo treinado em dados desbalanceados denota o custo zero dessa análise.

Quando acoplado ao classificador de regressão logística, o SMOTE apresenta o melhor custo-benefício, demonstrando um salto de desempenho considerável em relação ao modelo treinado sem aplicação de qualquer método de balanceamento, ao mesmo tempo em que

apresenta o menor custo computacional entre os métodos de balanceamento.

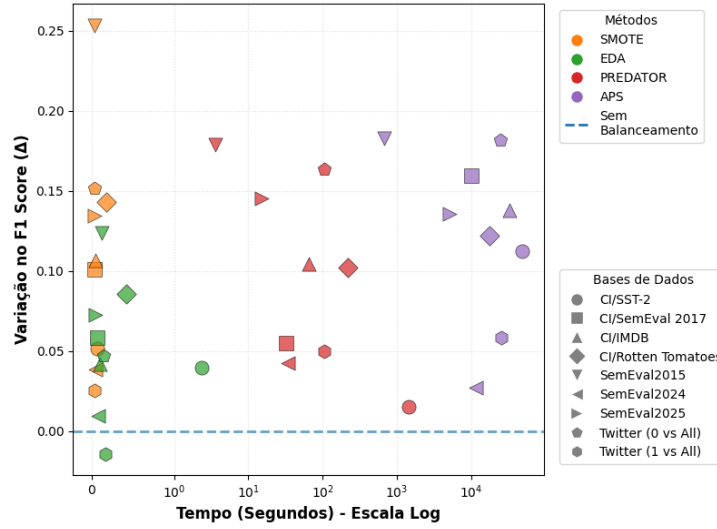


Figura 10: *Trade-off* desempenho preditivo vs. tempo em log segundos (LR).

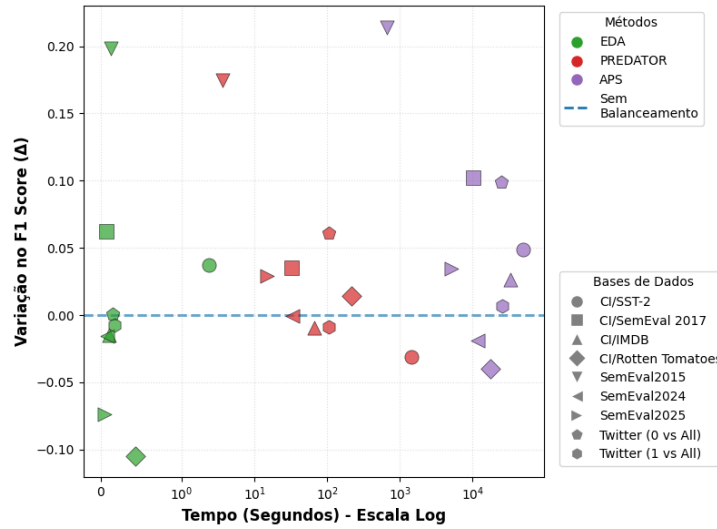


Figura 11: *Trade-off* desempenho preditivo vs. tempo em log segundos (XLM-R).

Apesar de o custo computacional de geração ser elevado no método APS em virtude do uso da LLM, é fundamental ressaltar que o processo de balanceamento de classes é uma etapa de pré-processamento executada de forma *offline* (Bayer; Kaufhold; Reuter, 2022). Uma vez que os dados sintéticos são gerados e o classificador final é treinado, a latência de inferência em produção dependerá exclusivamente do modelo de classificação (LR ou XLM-R), viabilizando o uso do método proposto em cenários reais onde a fase de treinamento possui maior flexibilidade de tempo e recursos.

6.5 Sumarização dos Resultados

A Tabela 18 apresenta qual método e cenário de balanceamento atingiu o melhor desempenho médio para cada base avaliada. As primeiras duas colunas apresentam o método que atingiu o melhor desempenho médio entre os métodos considerando os classificadores LR e XLM-R, respectivamente. A terceira coluna representa o método que apresentou o melhor desempenho médio entre todos os classificadores e cenários de desbalanceamento ($\alpha = 0,5$ a $\alpha = 1,0$). O método vencedor é reportado em conjunto com o classificador e cenário de balanceamento. O método proposto neste trabalho atinge o melhor desempenho geral em cinco das nove bases testadas.

Tabela 18: Resumo dos melhores resultados por base de dados.

Base	Melhor Resultado LR	Melhor Resultado XLM-R	Melhor Resultado Geral
CI/SST-2	APS ($\alpha = 1,0$)	APS ($\alpha = 1,0$)	XLM-R: APS ($\alpha = 1,0$)
CI/SemEval 2017	APS ($\alpha = 1,0$)	APS ($\alpha = 0,5$)	XLM-R: APS ($\alpha = 0,5$)
CI/IMDB	SMOTE ($\alpha = 0,5$)	APS ($\alpha = 0,5$)	XLM-R: APS ($\alpha = 0,5$)
CI/Rotten Tomatoes	SMOTE ($\alpha = 0,5$)	PREDATOR ($\alpha = 0,5$)	LR: SMOTE ($\alpha = 0,5$)
SemEval 2015	SMOTE ($\alpha = 1,0$)	PREDATOR ($\alpha = 0,5$)	LR: SMOTE ($\alpha = 1,0$)
SemEval 2024	SMOTE ($\alpha = 0,5$)	APS ($\alpha = 0,5$)	LR: SMOTE ($\alpha = 0,5$)
SemEval 2025	PREDATOR ($\alpha = 1,0$)	PREDATOR ($\alpha = 0,5$)	XLM-R: PREDATOR ($\alpha = 0,5$)
Twitter Topics (0 vs All)	SMOTE ($\alpha = 0,5$)	APS ($\alpha = 1,0$)	XLM-R: APS ($\alpha = 1,0$)
Twitter Topics (1 vs All)	SMOTE ($\alpha = 0,5$)	APS ($\alpha = 1,0$)	XLM-R: APS ($\alpha = 1,0$)

As análises de significância estatística *post-hoc* ilustradas na Figura 8 confirmam que o método proposto atinge um desempenho significativamente superior quando combinado com regressão logística e comparado ao cenário desbalanceado. Contudo, o método proposto não apresenta diferenças significativas em relação ao SMOTE quando combinado à regressão logística, método tradicional de classificação.

Sob a perspectiva da qualidade textual gerada, a análise dos exemplos gerados revela que o método APS se destaca pela fluidez linguística e pela capacidade de induzir variância estrutural (*NED*), sem sofrer do colapso semântico observado no PREDATOR (BERTScore = $-0,06$). Contudo, enquanto o seletor heurístico assegura a preservação da polaridade e do estilo, a paráfrase pode suprimir orações secundárias em certos cenários, resultando em uma ligeira perda de granularidade (BERTScore = $0,26$).

Com base nessas evidências, pode-se responder à segunda questão de pesquisa (**QP2**): A utilização de estratégias de filtragem pode tornar o aumento de dados baseado em paráfrases eficaz na mitigação do desbalanceamento de classes. No entanto, os resultados das Figuras 10 e 11 trazem um ponto a ser considerado: o método APS aumenta acentuadamente o custo computacional da estratégia de classificação. Esse resultado evidencia

um *trade-off* claro entre o ganho de desempenho e o tempo de processamento. Apesar de o custo computacional de geração ser elevado no método APS em virtude do uso da LLM, é fundamental ressaltar que esta é uma etapa de pré-processamento executada de forma *offline* (Bayer; Kaufhold; Reuter, 2022). Uma vez que os dados sintéticos são gerados e o classificador final é treinado, a latência de inferência em produção dependerá exclusivamente do modelo de classificação (LR ou XLM-R), viabilizando o uso do método proposto em cenários reais onde a fase de treinamento possui maior flexibilidade de tempo e recursos.

7 Conclusões e Trabalhos Futuros

Este capítulo estrutura o encerramento do trabalho. A Seção 7.1 sintetiza as principais conclusões obtidas a partir dos resultados experimentais. Em seguida, a Seção 7.2 discute as limitações metodológicas e computacionais identificadas ao longo da pesquisa. Por fim, a Seção 7.3 delinea as direções e propostas para trabalhos futuros.

7.1 Conclusões

O desbalanceamento de classes é um desafio prevalente na tarefa de classificação de textos, uma vez que a predominância de instâncias da classe majoritária induz os modelos de aprendizado de máquina ao viés de predição, degradando o desempenho preditivo sobre a classe de interesse (Gao *et al.*, 2026). Embora estratégias como PREDATOR e paráfrase gerada por LLMs tenham sido propostas para tratar esse problema, identificou-se uma lacuna na avaliação destes métodos em comparação a abordagens tradicionais como SMOTE e EDA.

Como contribuição inicial diante desse contexto, esta dissertação investigou o impacto do uso de LLMs no cenário de desbalanceamento de dados para a classificação de textos. Para viabilizar essa avaliação, foram criadas quatro bases de dados desbalanceadas a partir da subamostragem controlada das bases originais. A disponibilização dessas bases como cenários de referência para estudos futuros é uma outra contribuição desse trabalho. O estudo preliminar revelou que o uso direto de paráfrases geradas por LLMs sem um mecanismo de filtragem não garante melhorias preditivas consistentes. Nesses cenários não filtrados, observou-se que a geração sintética por vezes introduzia redundância ou ruído semântico, chegando a degradar o desempenho preditivo do classificador em comparação ao *baseline* sem balanceamento e a métodos consolidados da literatura. Esse resultado evidencia que o uso direto de LLMs para sobreamostragem não se mostrou superior aos métodos tradicionais (QP1).

A partir do diagnóstico de que a geração direta de paráfrases apresenta limitações que

anulam os ganhos preditivos, propôs-se o método Aumento por Paráfrases Seleccionadas (APS) como principal contribuição do trabalho. Esse método de balanceamento consiste na geração de paráfrases por meio de uma LLM seguida de uma seleção baseada em duas vertentes: proximidade ao vetor protótipo e entropia de Shannon. Os resultados experimentais demonstram que o APS cumpre o seu propósito principal ao alcançar o desempenho preditivo superior na maioria das bases de dados testadas, atingindo a melhor posição de ranking médio na maioria dos cenários avaliados. A validação estatística confirmou que o método traz vantagens significativas em relação ao modelo de regressão logística treinado sem balanceamento. Contudo, o método apresenta um comportamento equivalente ao tradicional SMOTE nesse mesmo cenário. A partir desses resultados, demonstra-se que a estratégia de filtragem é determinante para o desempenho do método, comprovando que a inclusão de critérios de seleção permitiu ao APS superar as limitações da paráfrase ingênua (sem critérios de filtragem) e do PREDATOR (QP2).

7.2 Limitações

A base SemEval 2015 é composta por uma quantidade reduzida de exemplos (321 ao todo). Quando aplicada no protocolo de validação cruzada com 10 *folds*, o conjunto de testes consiste em apenas 32–33 exemplos, dentre os quais 4–5 exemplos pertencem à classe minoritária. Por esse motivo, o desvio padrão do desempenho preditivo obtido com essa base é maior que nas demais bases. Essa instabilidade limita as conclusões que podem ser inferidas a partir dos resultados obtidos nessa base.

Apesar dos resultados positivos que o método apresenta, a seleção por entropia do método APS utiliza a fronteira de decisão de um modelo com possíveis vieses, devido ao desbalanceamento da base usada para o seu treinamento. Nesse sentido, a entropia estimada por um modelo enviesado não necessariamente reflete a dificuldade real de classificação, podendo capturar artefatos do próprio desbalanceamento. Além disso, a confiabilidade da incerteza estimada pelo *probe* depende diretamente da calibração de suas probabilidades preditivas, aspecto que não foi formalmente avaliado neste estudo.

Embora [Demšar \(2006\)](#) não estabeleça um número mínimo de bases de dados para a aplicação do teste de Friedman e do teste *post-hoc* de Nemenyi, a utilização de nove bases constitui uma amostra relativamente limitada de conjuntos de dados. Essa limitação pode reduzir o poder estatístico do procedimento, tornando mais difícil rejeitar a hipótese nula.

Por fim, a análise experimental confirma uma limitação importante do APS: sua

aplicação gera um aumento no custo computacional. Esse panorama consolida um *trade-off* claro para o trabalho, evidenciando que o ganho em desempenho preditivo exige uma contrapartida em tempo de processamento, embora este custo fique restrito à etapa de preparação dos dados, sem penalizar a latência de inferência do classificador final.

7.3 Trabalhos Futuros

A partir das conclusões e das limitações identificadas nesta pesquisa, derivam-se algumas direções promissoras para investigações futuras:

- **Otimização do Custo Computacional:** Desenvolver estratégias para mitigar o elevado custo computacional do APS, explorando o uso de LLMs altamente otimizados em termos de parâmetros (como modelos de 1B a 3B parâmetros) ou técnicas de quantização, avaliando o impacto dessa redução no desempenho preditivo.
- **Avaliação com Classificadores e Arquiteturas Modernas:** Estender a validação experimental a classificadores baseados em arquiteturas *encoder* mais recentes, analisando a capacidade de generalização do método perante modelos que demonstram maior desempenho em *benchmarks* de PLN (He; Gao; Chen, 2023; Warner *et al.*, 2024).
- **Estratégias Alternativas de Amostragem:** Investigar diferentes políticas de amostragem a partir dos escores de $M(x)$, explorando abordagens como seleção estocástica ponderada, limiares adaptativos ou amostragem com restrição de diversidade, em vez de depender exclusivamente de um corte determinístico (*top-k*).
- **Avaliação com Múltiplos *Prompts*:** Investigar o comportamento do método a partir de *prompts* alternativos, mapeando como variações nas diretrizes da instrução impactam o padrão de paráfrase gerado pela LLM e a eficácia da seleção sintética.
- **Cenário Multiclasse:** Expandir e avaliar o método proposto em cenários de classificação multiclasse ($|Y| > 2$) e com desbalanceamento severo, mapeando como os mecanismos de seleção se comportam quando a fronteira de decisão envolve mais de duas classes.
- **Sensibilidade e Impacto da Taxa de Balanceamento (α):** Conduzir um estudo sistemático sobre o hiperparâmetro α , que controla o volume de exemplos

sintéticos inseridos na classe minoritária. Essa análise permitirá investigar a sensibilidade dos métodos a diferentes graus de sobreamostragem, mapeando seu impacto direto na estabilidade e eficácia da tarefa final de classificação.

7.4 Considerações Éticas

O desenvolvimento e a aplicação de métodos baseados em LLMs para geração de dados sintéticos demandam uma reflexão crítica acerca de suas implicações éticas e operacionais:

- **Propagação de Vieses e Justiça Algorítmica:** LLMs pré-treinadas absorvem correlações espúrias e vieses socioculturais presentes em seus dados de treinamento (Bender *et al.*, 2021). No contexto de aumento de dados, existe o risco inerente de que esses vieses sejam incorporados às paráfrases geradas e subsequentemente assimilados pelos classificadores supervisionados, impactando a equidade das decisões preditivas.
- **Impacto Ambiental e Eficiência Computacional:** Em consonância com as diretrizes de *Green AI*, reconhece-se que a inferência iterativa de LLMs para a geração massiva de dados impõe uma demanda energética e computacional relevante (Schwartz *et al.*, 2020). O *trade-off* entre o ganho marginal de desempenho preditivo e o custo em recursos computacionais deve ser ponderado na adoção prática do método.

REFERÊNCIAS

ABONIZIO, Hugo Queiroz; JUNIOR, Sylvio Barbon. Pre-trained Data Augmentation for Text Classification. *In: BRACIS 2020*. Rio Grande, Brazil: Springer-Verlag, 2020. p. 551–565. ISBN 978-3-030-61376-1. DOI: [10.1007/978-3-030-61377-8_38](https://doi.org/10.1007/978-3-030-61377-8_38). Disponível em: https://doi.org/10.1007/978-3-030-61377-8_38.

ABONIZIO, Hugo Queiroz; PARAISO, Emerson Cabrera; BARBON, Sylvio. Toward Text Data Augmentation for Sentiment Analysis. **IEEE TAI**, v. 3, p. 657–668, 5 out. 2022. ISSN 2691-4581. DOI: [10.1109/TAI.2021.3114390](https://doi.org/10.1109/TAI.2021.3114390). Disponível em: <https://ieeexplore.ieee.org/document/9543519/>.

ANABY-TAVOR, Ateret *et al.* Do not have enough data? Deep learning to the rescue! **Proc. Conf. AAAI Artif. Intell.**, Association for the Advancement of Artificial Intelligence (AAAI), v. 34, n. 05, p. 7383–7390, abr. 2020.

ANTYPAS, Dimosthenis *et al.* Twitter Topic Classification. *In: 29TH COLING*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, out. 2022. p. 3386–3400. Disponível em: <https://aclanthology.org/2022.coling-1.299/>.

BAYER, Markus; KAUFHOLD, Marc-André; REUTER, Christian. A Survey on Data Augmentation for Text Classification. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 7, dez. 2022. ISSN 0360-0300. DOI: [10.1145/3544558](https://doi.org/10.1145/3544558). Disponível em: <https://doi.org/10.1145/3544558>.

BENDER, Emily M. *et al.* On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . *In: PROCEEDINGS of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Virtual Event, Canada: Association for Computing Machinery, 2021. (FAccT '21), p. 610–623. ISBN 9781450383097. DOI: [10.1145/3442188.3445922](https://doi.org/10.1145/3442188.3445922). Disponível em: <https://doi.org/10.1145/3442188.3445922>.

BIEWALD, Lukas. **Experiment Tracking with Weights and Biases**. [*S. l.: s. n.*], 2020. Software available from wandb.com. Disponível em: <https://www.wandb.com/>.

BROWN, Tom B. *et al.* Language models are few-shot learners. *In: PROCEEDINGS of the 34th International Conference on Neural Information Processing Systems*.

Vancouver, BC, Canada: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

CHAWLA, Nitesh V. *et al.* SMOTE: synthetic minority over-sampling technique. **J. Artif. Int. Res.**, AI Access Foundation, El Segundo, CA, USA, v. 16, n. 1, p. 321–357, jun. 2002. ISSN 1076-9757.

CONNEAU, Alexis *et al.* Unsupervised Cross-lingual Representation Learning at Scale. *In: JURAFSKY, Dan et al. (ed.). Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, jul. 2020. p. 8440–8451. DOI: [10.18653/v1/2020.acl-main.747](https://doi.org/10.18653/v1/2020.acl-main.747). Disponível em: <https://aclanthology.org/2020.acl-main.747/>.

CUNHA, Washington; ROCHA, Leonardo; GONÇALVES, Marcos André. A thorough benchmark of automatic text classification: From traditional approaches to large language models. **arXiv preprint arXiv:2504.01930**, 2025.

DEMŠAR, Janez. Statistical Comparisons of Classifiers over Multiple Data Sets. **J. Mach. Learn. Res.**, JMLR.org, v. 7, 2006.

DEVLIN, Jacob *et al.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In: BURSTEIN, Jill; DORAN, Christy; SOLORIO, Thamar (ed.). 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, jun. 2019. p. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). Disponível em: <https://aclanthology.org/N19-1423/>.

DIMITROV, Dimitar *et al.* SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. *In: PROCEEDINGS of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Mexico City, Mexico: Association for Computational Linguistics, jun. 2024. p. 2009–2026. DOI: [10.18653/v1/2024.semeval-1.275](https://doi.org/10.18653/v1/2024.semeval-1.275). Disponível em: <https://aclanthology.org/2024.semeval-1.275/>.

EDUNOV, Sergey *et al.* Understanding Back-Translation at Scale. *In: RILOFF, Ellen et al. (ed.). Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, out. 2018. p. 489–500. DOI: [10.18653/v1/D18-1045](https://doi.org/10.18653/v1/D18-1045). Disponível em: <https://aclanthology.org/D18-1045/>.

FONSECA, Guilherme *et al.* Instance-selection-inspired undersampling strategies for bias reduction in small and large language models for binary text classification. *In: ACL. [S. l.: s. n.]*, 2025. p. 9323–9340.

GAO, Xinyi *et al.* A comprehensive survey on imbalanced data learning. **Frontiers of Computer Science**, v. 20, n. 11, p. 2011622, mar. 2026. ISSN 2095-2236. DOI: [10.1007/s11704-025-50274-7](https://doi.org/10.1007/s11704-025-50274-7). Disponível em: <https://doi.org/10.1007/s11704-025-50274-7>.

GHOSH, Aniruddha *et al.* SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In: NAKOV, Preslav *et al.* (ed.). **SemEval 2015**. Denver, Colorado: Association for Computational Linguistics, jun. 2015. p. 470–478. DOI: [10.18653/v1/S15-2080](https://aclanthology.org/S15-2080/). Disponível em: <https://aclanthology.org/S15-2080/>.

GRATTAFIORI, Aaron *et al.* **The Llama 3 Herd of Models**. [*S. l.: s. n.*], 2024. arXiv: [2407.21783](https://arxiv.org/abs/2407.21783) [*cs.AI*]. Disponível em: <https://arxiv.org/abs/2407.21783>.

HE, Haibo; GARCIA, Edwardo A. Learning from Imbalanced Data. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 9, p. 1263–1284, 2009. DOI: [10.1109/TKDE.2008.239](https://doi.org/10.1109/TKDE.2008.239).

HE, Pengcheng; GAO, Jianfeng; CHEN, Weizhu. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In: THE Eleventh International Conference on Learning Representations (ICLR). [*S. l.: s. n.*], 2023.

JANG, Joel *et al.* Sequential targeting: A continual learning approach for data imbalance in text classification. **Expert Systems with Applications**, v. 179, p. 115067, 2021. ISSN 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2021.115067>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S095741742100508X>.

JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models**. 3rd. [*S. l.*]: Online, 2026. Online manuscript released January 6, 2026. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>.

LEVENSHTEIN, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. **Soviet Physics Doklady**, v. 10, n. 8, p. 707–710, 1966.

LEWIS, David D.; GALE, William A. **A Sequential Algorithm for Training Text Classifiers**. [*S. l.: s. n.*], 1994. arXiv: [cmp-lg/9407020](https://arxiv.org/abs/cmp-lg/9407020) [*cmp-lg*]. Disponível em: <https://arxiv.org/abs/cmp-lg/9407020>.

LIU, Yinhan *et al.* **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. [*S. l.: s. n.*], 2019. arXiv: [1907.11692](https://arxiv.org/abs/1907.11692) [*cs.CL*]. Disponível em: <https://arxiv.org/abs/1907.11692>.

MAAS, Andrew L. *et al.* Learning Word Vectors for Sentiment Analysis. *In*: 49TH ACL. Portland, Oregon, USA: Association for Computational Linguistics, jun. 2011. p. 142–150. Disponível em: <http://www.aclweb.org/anthology/P11-1015>.

MANNING, Christopher D; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **Introduction to Information Retrieval**. [S. l.]: Cambridge University Press, 2008.

MIKOLOV, Tomas *et al.* Distributed Representations of Words and Phrases and their Compositionality. *In*: BURGESS, C.J. *et al.* (ed.). **Advances in Neural Information Processing Systems**. [S. l.]: Curran Associates, Inc., 2013. v. 26. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

MIN, Bonan *et al.* Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. **ACM CSUR**, Association for Computing Machinery, New York, NY, USA, v. 56, n. 2, set. 2023. ISSN 0360-0300. DOI: [10.1145/3605943](https://doi-org.ez24.periodicos.capes.gov.br/10.1145/3605943). Disponível em: <https://doi-org.ez24.periodicos.capes.gov.br/10.1145/3605943>.

MINAEE, Shervin *et al.* Deep Learning-based Text Classification: A Comprehensive Review. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 54, n. 3, abr. 2021. ISSN 0360-0300. DOI: [10.1145/3439726](https://doi.org/10.1145/3439726). Disponível em: <https://doi.org/10.1145/3439726>.

MUHAMMAD, Shamsuddeen Hassan *et al.* SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection. *In*: SEMEVAL-2025. [S. l.: s. n.], 2025.

NATARAJAN, Nagarajan *et al.* Learning with noisy labels. **Advances in Neural Information Processing Systems (NeurIPS)**, v. 26, p. 1196–1204, 2013.

PANG, Bo; LEE, Lillian. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *In*: ACL. [S. l.: s. n.], 2005.

PATIL, Annapurna P *et al.* T5W: A Paraphrasing Approach to Oversampling for Imbalanced Text Classification. *In*: CONECCT. [S. l.: s. n.], 2022. p. 1–6. DOI: [10.1109/CONECCT55679.2022.9865812](https://doi.org/10.1109/CONECCT55679.2022.9865812).

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine Learning in Python. **J. Mach. Learn. Res.**, JMLR.org, v. 12, null, p. 2825–2830, nov. 2011. ISSN 1532-4435.

PELLICER, Lucas Francisco Amaral Orosco; FERREIRA, Taynan Maier; COSTA, Anna Helena Reali. Data augmentation techniques in natural language processing. **Appl. Soft Comput.**, v. 132, p. 109803, 2023. DOI:

[10.1016/J.ASOC.2022.109803](https://doi.org/10.1016/j.asoc.2022.109803). Disponível em:
<https://doi.org/10.1016/j.asoc.2022.109803>.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher. GloVe: Global Vectors for Word Representation. *In*: MOSCHITTI, Alessandro; PANG, Bo; DAELEMANS, Walter (ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, out. 2014. p. 1532–1543. DOI: [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162). Disponível em: <https://aclanthology.org/D14-1162/>.

PETERS, Matthew E. *et al.* Deep Contextualized Word Representations. *In*: WALKER, Marilyn; JI, Heng; STENT, Amanda (ed.). **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**. New Orleans, Louisiana: Association for Computational Linguistics, jun. 2018. p. 2227–2237. DOI: [10.18653/v1/N18-1202](https://doi.org/10.18653/v1/N18-1202). Disponível em: <https://aclanthology.org/N18-1202/>.

RADLIŃSKI, Łukasz; GUŚCIORA, Mateusz; KOCON, Jan. Backtranslation and Paraphrasing in the LLM Era? Comparing Data Augmentation Methods for Emotion Classification. *In*: LNCS. [S. l.]: Springer Science e Business Media Deutschland GmbH, 2025. 15903 LNCS, p. 3–17. ISBN 9783031976254. DOI: [10.1007/978-3-031-97626-1_1](https://doi.org/10.1007/978-3-031-97626-1_1).

RAFFEL, Colin *et al.* Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **Journal of Machine Learning Research**, v. 21, n. 140, p. 1–67, 2020. Disponível em: <http://jmlr.org/papers/v21/20-074.html>.

ROSENTHAL, Sara; FARRA, Noura; NAKOV, Preslav. SemEval-2017 Task 4: Sentiment Analysis in Twitter. *In*: BETHARD, Steven *et al.* (ed.). **SemEval-2017**. Vancouver, Canada: ACL, ago. 2017. p. 502–518. DOI: [10.18653/v1/S17-2088](https://doi.org/10.18653/v1/S17-2088). Disponível em: <https://aclanthology.org/S17-2088/>.

SANH, Victor *et al.* DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. **ArXiv**, abs/1910.01108, 2019.

SCHWARTZ, Roy *et al.* Green AI. **Communications of the ACM**, ACM New York, NY, USA, v. 63, n. 12, p. 54–63, 2020.

SETTLES, Burr. **Active Learning Literature Survey**. [S. l.], 2009.

SNELL, Jake; SWERSKY, Kevin; ZEMEL, Richard S. **Prototypical Networks for Few-shot Learning**. [S. l.: s. n.], 2017. arXiv: [1703.05175](https://arxiv.org/abs/1703.05175) [cs.LG]. Disponível em: <https://arxiv.org/abs/1703.05175>.

SOCHER, Richard *et al.* Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *In: EMNLP*. Seattle, Washington, USA: Association for Computational Linguistics, out. 2013. p. 1631–1642. Disponível em: <https://www.aclweb.org/anthology/D13-1170>.

VASWANI, Ashish *et al.* Attention is All you Need. *In: GUYON, I. et al. (ed.). Advances in Neural Information Processing Systems*. [S. l.]: Curran Associates, Inc., 2017. v. 30. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

WARNER, Benjamin *et al.* Smarter, Better, Faster, Longer: A Modern BERT Architecture. **arXiv preprint arXiv:2412.13663**, 2024.

WEI, Jason W.; ZOU, Kai. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *In: INUI, Kentaro et al. (ed.). EMNLP-IJCNLP*. [S. l.]: Association for Computational Linguistics, 2019. p. 6381–6387. DOI: [10.18653/V1/D19-1670](https://doi.org/10.18653/v1/D19-1670). Disponível em: <https://doi.org/10.18653/v1/D19-1670>.

WILCOXON, Frank. Individual Comparisons by Ranking Methods. **Biometrics Bulletin**, JSTOR, v. 1, n. 6, p. 80–83, 1945. DOI: [10.2307/3001968](https://doi.org/10.2307/3001968).

WOLF, Thomas *et al.* **HuggingFace’s Transformers: State-of-the-art Natural Language Processing**. [S. l.: s. n.], 2020. arXiv: [1910.03771](https://arxiv.org/abs/1910.03771) [cs.CL]. Disponível em: <https://arxiv.org/abs/1910.03771>.

WU, Xing *et al.* Conditional BERT Contextual Augmentation. *In: RODRIGUES, João M. F. et al. (ed.). Computational Science – ICCS 2019*. Cham: Springer International Publishing, 2019. p. 84–95. ISBN 978-3-030-22747-0.

ZHANG, Tianyi *et al.* **BERTScore: Evaluating Text Generation with BERT**. [S. l.: s. n.], 2020. arXiv: [1904.09675](https://arxiv.org/abs/1904.09675) [cs.CL]. Disponível em: <https://arxiv.org/abs/1904.09675>.