UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOSÉ VALDIR SANTIAGO NETO

COMPARAÇÃO DE ESTRATÉGIAS DE PROMPTING EM LLMS PARA CLASSIFICAÇÃO DE TEXTO EM PORTUGUÊS: UM ESTUDO EMPÍRICO ENTRE ZERO-SHOT, FEW-SHOT E CHAIN-OF-THOUGHT

RUSSAS
2026

JOSÉ VALDIR SANTIAGO NETO

COMPARAÇÃO DE ESTRATÉGIAS DE PROMPTING EM LLMS PARA CLASSIFICAÇÃO DE TEXTO EM PORTUGUÊS: UM ESTUDO EMPÍRICO ENTRE ZERO-SHOT, FEW-SHOT E CHAIN-OF-THOUGHT

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação Da Universidade Federal do Ceará como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Pedro Helton Magalhães Pinheiro

RUSSAS

2026

JOSÉ VALDIR SANTIAGO NETO

COMPARAÇÃO DE ESTRATÉGIAS DE PROMPTING EM LLMS PARA CLASSIFICAÇÃO DE TEXTO EM PORTUGUÊS: UM ESTUDO EMPÍRICO ENTRE ZERO-SHOT, FEW-SHOT E CHAIN-OF-THOUGHT

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em: 20/01/2026.

BANCA EXAMINADORA

_____

Prof. Dr. Pedro Helton Magalhães Pinheiro
(Orientador)
Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Edvan Cordeiro de Miranda
Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Reuber Regis de Melo
Universidade Federal do Ceará (UFC)

A Deus.

A minha mãe, Natália.

AGRADECIMENTOS

Ao Prof. Dr. Pedro Helton Magalhães Pinheiro, pela excelente orientação.

Aos professores participantes da banca examinadora Prof. Dr. Edvan Cordeiro de Miranda e Prof. Dr. Reuber Regis de Melo pelo tempo, pelas valiosas colaborações e sugestões.

RESUMO

A área de Processamento de Linguagem Natural (PLN) foi profundamente impactada pelo surgimento dos Grandes Modelos de Linguagem (LLMs), que introduziram o paradigma de aprendizado no contexto (*in-context learning*). Essa abordagem permite realizar tarefas complexas via instruções (*prompts*), reduzindo a necessidade de treinamento supervisionado tradicional. No entanto, para tarefas específicas em português, ainda há dúvidas sobre quando essas novas estratégias justificam seu custo computacional frente a métodos clássicos já consolidados. Este trabalho investiga esse cenário ao comparar empiricamente diferentes estratégias de *prompting* — incluindo *zero-shot* (via inferência textual e *embeddings*), *few-shot* e *Chain-of-Thought* (CoT) — na tarefa de classificação de sentimentos. Utilizando um conjunto de dados sintético e balanceado, o estudo avaliou não apenas a eficácia (Macro-F1), mas também a latência em CPU, contrastando os LLMs com um *baseline* supervisionado (TF-IDF + SVM). Os resultados indicam que, em dados com pistas léxicas claras, as técnicas clássicas ainda oferecem desempenho superior e máxima eficiência. Entre as abordagens sem supervisão, o uso de inferência em linguagem natural (NLI) mostrou-se a alternativa mais robusta em qualidade, enquanto a estratégia baseada em *embeddings* destacou-se pela velocidade. As abordagens gerativas *few-shot*, por sua vez, revelaram desafios de estabilidade na adesão aos rótulos rígidos. Conclui-se que a adoção de LLMs deve ser criteriosa, ponderando a disponibilidade de dados rotulados e as restrições de infraestrutura.

Palavras-chave: engenharia de *prompts*; modelos de linguagem; *zero-shot*; *few-shot*; classificação de sentimentos.

ABSTRACT

The field of Natural Language Processing (NLP) has been profoundly impacted by the emergence of Large Language Models (LLMs), which introduced the in-context learning paradigm. This approach allows for performing complex tasks via prompts, reducing the need for traditional supervised training. However, for specific tasks in Portuguese, questions remain regarding when these new strategies justify their computational cost compared to established classical methods. This work investigates this scenario by empirically comparing different prompting strategies — including zero-shot (via textual inference and embeddings), few-shot, and Chain-of-Thought (CoT) — in the sentiment classification task. Using a balanced synthetic dataset, the study evaluated not only efficacy (Macro-F1) but also CPU latency, benchmarking LLMs against a supervised baseline (TF-IDF + SVM). Results indicate that, for data with clear lexical cues, classical techniques still offer superior performance and maximum efficiency. Among unsupervised approaches, the use of Natural Language Inference (NLI) proved to be the most robust alternative in terms of quality, while the embeddings-based strategy stood out for its speed. Generative few-shot approaches, in turn, revealed stability challenges in adhering to rigid labels. It is concluded that the adoption of LLMs must be judicious, weighing the availability of labeled data against infrastructure constraints.

Keywords: prompt engineering; large language models; zero-shot; few-shot; sentiment classification.

SUMÁRIO

1 INTRODUÇÃO

Historicamente, a resolução de problemas em Processamento de Linguagem Natural (PLN), como a classificação de sentimentos, fundamentava-se predominantemente no paradigma do aprendizado supervisionado. Essa abordagem tradicional impõe barreiras significativas ao desenvolvimento ágil: exige a construção prévia de vastos conjuntos de dados rotulados (*labeled datasets*), processo que demanda esforço humano intensivo para a anotação manual, eleva substancialmente os custos operacionais e dilata os prazos de implementação. Além disso, tais modelos frequentemente requerem treinamento específico para cada nova tarefa, dificultando a generalização rápida para novos domínios.

Entretanto, o advento dos Grandes Modelos de Linguagem (LLMs) promoveu uma ruptura nessa dinâmica. Ao serem pré-treinados em escalas massivas de texto, esses modelos adquiriram a capacidade de realizar tarefas complexas por meio de instruções em linguagem natural (*prompts*), mitigando a necessidade de treinamento do zero e a dependência estrita de grandes volumes de dados anotados. Essa transição para o aprendizado no contexto (*in-context learning*) torna-se atraente por prometer resultados imediatos e redução de custos de curadoria. Contudo, surge um dilema prático para a engenharia de sistemas: deve-se confiar na inferência direta (*zero-shot*), investir na seleção de exemplos demonstrativos (*few-shot*) ou induzir o raciocínio passo a passo (*Chain-of-Thought*)?

Este trabalho nasce dessa tensão e compara, de modo sistemático, essas estratégias com um baseline clássico e estável (TF-IDF + SVM), tornando explícitos os *trade-offs* entre custo e qualidade. Em termos diretos: quando o *prompting* é suficiente e quando o "arroz com feijão" supervisionado ainda entrega mais pelo que custa.

A pergunta de pesquisa é objetiva e orientada à aplicação: como diferentes estratégias de *prompting* afetam o desempenho de grandes modelos de linguagem na tarefa de classificação de textos em português (PT-BR)? Interessa-nos evidência, e não impressão. Por isso, o objetivo geral é comparar empiricamente essas abordagens em um cenário de análise de sentimento com mensuração conjunta de desempenho e de custo operacional.

Para responder a essa questão, adotamos um percurso reprodutível e enxuto. Definimos um conjunto de dados sintético e balanceado com três classes (negativo, neutro e positivo), a fim de isolar o efeito das estratégias. Utilizamos divisão estratificada 70/15/15, com semente fixa, e construímos *prompts* padronizados com variações controladas (ordem e quantidade de exemplos, janela de contexto, idioma e estilo das instruções), mantendo a saída rigidamente restrita aos rótulos.

A avaliação prioriza Macro-F1 e, quando pertinente, inclui PR-AUC e análise de matriz de confusão para identificar padrões de erro. Em paralelo, estimamos latência e custo (tempo médio por amostra e, quando aplicável, contagem de *tokens*), registrando todas as métricas e figuras para facilitar a reprodutibilidade. A comparação envolve um *baseline* supervisionado (TF-IDF + SVM linear) e quatro trilhas de *prompting*: (i) *zero-shot* ancorado em inferência textual com template declarativo; (ii) *few-shot* gerativo com rótulos fechados; (iii) *few-shot* com *CoT*, preservando a resposta final restrita; e (iv) *zero-shot* por *embeddings* multilíngues (MiniLM), medindo similaridade do cosseno entre textos e descrições de classe. Toda a execução ocorre em CPU, discutida como potencial ameaça à validade externa.

A relevância do estudo está em oferecer um quadro decisório claro para quem precisa alocar tempo e recursos com responsabilidade. Ao colocar Macro-F1 ao lado de latência, torna-se possível reconhecer: (a) quando o *prompting* já entrega boa relação esforço–benefício; (b) quando *embeddings* em *zero-shot* são um atalho eficaz; e (c) quando o baseline tradicional permanece a opção mais segura. O resultado interessa à academia — pela sistematização do protocolo e pela nitidez dos recortes — e ao campo profissional, especialmente sob restrições reais de orçamento e prazo.

Quanto à organização: a Seção 1 apresenta o tema, delimita o problema, explicita a pergunta e integra os objetivos à introdução. A Seção 2 reúne o referencial teórico e os trabalhos relacionados (LLMs, estratégias de *prompting — zero-shot, few-shot* e *CoT —*, *NLI* e usos de *embeddings*). A Seção 3 detalha o desenho metodológico, os dados, os modelos, a construção dos *prompts*, o protocolo experimental e as métricas, incluindo o procedimento de estimativa de custo e latência. A Seção 4 apresenta e discute os resultados, destacando os principais *trade-offs* à luz dos números. Por fim, a Seção 5 traz as conclusões, reconhece limitações — como o uso de dados sintéticos e a execução em CPU — e aponta caminhos futuros, entre eles dados reais mais ruidosos, modelos instruídos para PT-BR, *self-consistency* e otimização sistemática de *prompts*.

Em suma, trata-se de um estudo empírico com foco prático: medir, comparar e esclarecer quando e por que cada estratégia faz mais sentido — especialmente quando o relógio e o orçamento pesam na decisão.

2 OBJETIVOS

Esta seção detalha o objetivo principal e os objetivos específicos delineados para conduzir este estudo comparativo.

2.1 Objetivo Geral

Comparar empiricamente o desempenho de diferentes estratégias de *prompting* (*zero-shot, few-shot* e *Chain-of-Thought*) em grandes modelos de linguagem na tarefa de classificação de sentimentos em português (PT-BR), contrastando-as com um *baseline* supervisionado clássico para explicitar os *trade-offs* entre custo operacional, latência e qualidade da classificação.

2.2 Objetivos Específicos

Para alcançar o objetivo geral, definiram-se as seguintes metas específicas:

- **Estabelecer um *baseline* de referência:** Implementar e avaliar um classificador supervisionado tradicional (TF-IDF + SVM linear) em um conjunto de dados sintético e balanceado, definindo um teto prático de desempenho para dados linearmente separáveis.
- **Avaliar estratégias de *prompting*:** Mensurar a eficácia de quatro abordagens distintas de inferência sem treino supervisionado: *zero-shot* via *NLI*, *few-shot* gerativo com saída restrita, *few-shot* com *Chain-of-Thought* (*CoT*) e *zero-shot* baseado em similaridade de *embeddings*.
- **Analisar a sensibilidade dos modelos:** Investigar como variações na construção do *prompt* — especificamente a quantidade de exemplos (k), a ordem de apresentação e o estilo da instrução — afetam a estabilidade e a acurácia das previsões.
- **Quantificar custo e eficiência:** Medir a latência média por amostra (em CPU) e o custo computacional de cada estratégia, correlacionando esses dados com a métrica de desempenho (Macro-F1).
- **Sistematizar critérios de decisão:** Oferecer um quadro comparativo que auxilie na escolha da estratégia mais adequada (supervisionada, baseada em *embeddings* ou gerativa) conforme as restrições de orçamento, tempo e disponibilidade de dados rotulados.

3 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta e discute as principais contribuições da literatura sobre grandes modelos de linguagem aplicados à classificação de textos em português, com foco em como o aprendizado no contexto (*in-context learning*) viabiliza o uso de exemplos diretamente no *prompt* e substitui, em parte, o treinamento tradicional. Partimos dos fundamentos de LLMs e do papel do contexto para, em seguida, analisar estratégias de *prompting* — *zero-shot*, *few-shot* e a inclusão de raciocínio passo a passo — destacando seus efeitos práticos em tarefas supervisionadas.

Exploramos ainda fatores de sensibilidade do *prompt*, como ordem e quantidade de exemplos, extensão do contexto e estilo das instruções, e examinamos técnicas complementares, como o uso de inferência textual e *embeddings* para cenários de *zero-shot*. Por fim, discutimos critérios de avaliação (com ênfase em Macro-F1, PR-AUC e matrizes de confusão) e a dimensão de custo e latência, situando esse debate no panorama específico do PT-BR — onde há avanços relevantes, mas também lacunas de reprodutibilidade e protocolos padronizados. O objetivo é contextualizar o estado da arte, identificar convergências e pontos em aberto, e oferecer o embasamento necessário para entender os desafios e as escolhas metodológicas que orientam este estudo.

3.1 LLMs e *In-Context Learning* (*ICL*)

No plano conceitual, LLMs são modelos probabilísticos pré-treinados para estimar a próxima unidade linguística a partir de grandes corpora; dessa competência geral emergem comportamentos úteis em múltiplas tarefas, mesmo sem novo treinamento específico. *In-Context Learning* (*ICL*) descreve precisamente esse efeito: o modelo ajusta a resposta condicionada ao próprio *prompt*, sem atualização de pesos. Em *zero-shot*, bastam a instrução e a definição dos rótulos para orientar a saída; em *few-shot*, um pequeno conjunto de exemplos (k instâncias) funciona como demonstração do padrão desejado, reforçando a indução no contexto (Sahoo et al., 2024; Dong et al., 2024; Liu et al., 2023/2024).

Do ponto de vista prático, o *ICL* favorece prototipagem rápida e transferência para domínios com poucos rótulos, reduzindo custos de curadoria e de treinamento supervisionado. Em contrapartida, a literatura relata limitações recorrentes: alta sensibilidade à redação do *prompt* (ordem e formulação dos exemplos, estilo da instrução), janela de contexto finita —

que restringe a quantidade de evidência acessível — e variabilidade entre modelos e idiomas, capaz de alterar de forma substancial o desempenho observado (Sahoo et al., 2024; Liu et al., 2023/2024). Tais fatores tornam o resultado dependente de escolhas textuais e operacionais aparentemente discretas, porém decisivas.

Em cenários aplicados — como classificação de sentimento em PT-BR —, recomenda-se calibrar o desenho do *prompt* à luz desses *trade-offs*: delimitar claramente o espaço de resposta (rótulos fechados), controlar a quantidade e a ordem dos exemplos e documentar sistematicamente configurações de contexto, latência e custo de inferência. Essa disciplina permite comparar *zero-shot* e *few-shot* de modo justo, identificar ganhos reais de *ICL* e estabelecer limites de validade externa diante de mudanças de domínio, distribuição ou idioma (Dong et al., 2024; Liu et al., 2023/2024).

3.2 Estratégias de *Prompting*

Há dois eixos principais: *prompts* instrucionais, com instruções claras, restrições de formato e critérios de validação da saída; e *prompts* exemplificativos, que ancoram o comportamento do modelo em poucos exemplos rotulados apresentados como demonstração do padrão desejado. Boas práticas incluem: especificar explicitamente o formato de resposta — “responda exatamente com {negativo, neutro, positivo}” —; preferir rótulos curtos e inequívocos; e usar templates de hipótese (*NLI*) para mapear rótulos a enunciados afirmativos, reduzindo ambiguidades e facilitando verificação automática. Em tarefas sensíveis a trechos literais, marcadores de verbatim evitam normalizações indevidas do texto de entrada. A variação sistemática de k no *few-shot* ($k \in \{2, 4, 8, 16\}$) revela a curva de ganho e pontos de retorno decrescente, orientando a escolha do menor k que sustenta o desempenho desejado.

Com contexto longo, o *many-shot* (k elevado) amplia a evidência fornecida ao modelo, mas enfrenta limites da janela de contexto e degradação de atenção, que podem neutralizar parte do benefício quando muitos exemplos competem por foco (Agarwal et al., 2024). A recomendação é balancear cobertura e concisão: priorizar exemplos prototípicos, manter instruções sucintas e preservar um espaço de resposta rigidamente fechado, garantindo calibragem e comparabilidade entre execuções. Assim, a escolha entre instrução pura, *few-shot* enxuto ou *many-shot* passa a ser uma decisão orientada por evidência e pelos limites efetivos de contexto do modelo.

3.3 *Chain-of-Thought (CoT)* e *Self-Consistency*

Chain-of-Thought (*CoT*) instrui o modelo a explicitar um raciocínio passo a passo antes da resposta final. A literatura indica ganhos sobretudo em tarefas de múltiplas etapas — cálculo, lógica, demonstrações — nas quais decompor o problema em subpartes ajuda a reduzir erros e ambiguidades (Wei et al., 2022). Em classificação com rótulos curtos e espaço de saída rigidamente fechado, porém, o benefício tende a ser limitado: quando a decisão é, em essência, um mapeamento texto → rótulo, o custo adicional de gerar cadeias de raciocínio raramente se traduz em melhoria substantiva de acurácia (Wei et al., 2022).

A variante *self-consistency* busca mitigar a variância gerando múltiplas cadeias de raciocínio independentes e agregando o veredito por votação. Esse procedimento costuma estabilizar as saídas e reduzir a sensibilidade a pequenas variações do *prompt*, especialmente quando há caminhos alternativos plausíveis que levam à mesma resposta (Wei et al., 2022). Em contrapartida, ele eleva latência e custo de inferência; assim, sua adoção é mais justificável quando a robustez pesa mais do que a restrição de recursos — e menos indicada em cenários nos quais orçamento computacional e tempo de resposta são prioridade.

3.4 Fatores de Sensibilidade: ordem, posição e tamanho do contexto

O desempenho em *ICL* responde de maneira aguda a três decisões aparentemente simples: a ordem dos exemplos no *few-shot*, a posição relativa da consulta dentro do *prompt* e o tamanho efetivo do contexto. Efeitos de primazia e recência fazem diferença: exemplos exibidos primeiro ou por último tendem a “puxar” o comportamento do modelo, enquanto itens soterrados no meio perdem influência. Em contextos muito longos, essa assimetria se acentua — achados recentes descrevem o fenômeno *lost in the middle*, em que evidências posicionadas na região central recebem menos atenção e, por consequência, degradam a estabilidade das previsões (Liu et al., 2024).

Em cenários aplicados, a mitigação passa por escolhas de layout no próprio *prompt*: ordenar exemplos por proximidade de domínio, agrupar por classe para reduzir alternâncias ruidosas e manter, fisicamente perto da consulta, tanto as instruções quanto a especificação do formato de saída. Em paralelo, controlar o comprimento do contexto — evitando alongamentos desnecessários de exemplos ou trechos redundantes — ajuda a preservar sinal útil e a reduzir a variância entre execuções. Em síntese, ao tratar ordem, posição e tamanho do

contexto como hiperparâmetros textuais (e não como detalhe de redação), aumenta-se a chance de o modelo aproveitar as evidências certas, no lugar certo, na hora certa.

3.5 Avaliação: métricas e custo

Para garantir a precisão analítica e a comparabilidade dos resultados, a avaliação de desempenho baseia-se em métricas consolidadas na literatura de aprendizado de máquina. Inicialmente, definem-se a Precisão (P) e a Revocação (R). A precisão mensura a confiabilidade das predições positivas, ou seja, a razão entre os verdadeiros positivos (TP) e o total de instâncias classificadas como positivas (TP+FP). A revocação, por sua vez, avalia a cobertura do modelo, calculada como a razão entre os verdadeiros positivos e o total de instâncias que realmente pertencem à classe (TP+FN). A partir dessas duas grandezas, obtém-se o F1-Score, que representa a média harmônica entre elas, penalizando valores extremos. As fórmulas são dadas por:

$$P = \frac{TP}{TP+FP},$$

$$R = \frac{TP}{TP+FN},$$

$$F1 = 2 \cdot \frac{P \cdot R}{P+R}$$

Para a agregação dos resultados em um cenário multiclasse, utiliza-se a Macro-F1, que consiste na média aritmética simples dos escores F1 calculados individualmente para cada classe (C), conforme a equação:

$$Macro\ F1 = \frac{1}{|C|} \sum_{i=1}^{|C|} F1_i$$

Nesse contexto, a escolha da Macro-F1 como métrica principal justifica-se pela necessidade de ponderar as classes de forma equilibrada, inclusive sob pequenos desequilíbrios. Ao calcular a média não ponderada, a métrica impede que categorias majoritárias dominem a leitura dos resultados, oferecendo um panorama justo da eficácia do modelo em todos os sentimentos avaliados. Em complemento, a análise individualizada da precisão e revocação por classe permite explicitar assimetrias de aprendizado que a média global poderia esconder.

Adicionalmente, emprega-se a PR-AUC (*Area Under the Precision-Recall Curve*). Esta métrica é particularmente informativa em situações de desbalanceamento relevante ou quando a distinção entre classes é sutil, pois captura a dinâmica de *trade-off* entre precisão e revocação ao longo de diferentes limiares de decisão, focando estritamente na qualidade da

detecção da classe de interesse. A matriz de confusão, por sua vez, oferece a visão operacional dos erros: ao cruzar rótulos reais e preditos, ela revela confusões sistemáticas entre vizinhos, identifica classes "ímã" que atraem previsões indevidas e mapeia zonas de baixa separabilidade, orientando intervenções concretas no desenho do *prompt*.

Do ponto de vista aplicado, a avaliação incorpora custo e latência como cidadãos de primeira classe. Em ambientes locais, mede-se o tempo médio por amostra estritamente em CPU; em cenários via API, o consumo de *tokens* de entrada e saída traduz diretamente o impacto financeiro e ajuda a calibrar escolhas de contexto. Essa dupla métrica — desempenho e custo — sustenta decisões realistas quando o objetivo é maximizar valor sob restrições de orçamento e prazo .

Por fim, em configurações *zero-shot*, medidas de confiança funcionam como camada de governança. Escores de *entailment* em NLI, ou a margem entre o rótulo vencedor e o segundo colocado, permitem identificar previsões incertas e acionar revisão humana ou recusa da automação . Integradas a um fluxo de *human-in-the-loop*, essas sinalizações reduzem risco operacional, garantindo que o sistema seja não só preciso, mas também sustentável na prática.

3.6 LLMs em Português

No contexto do português do Brasil, o ecossistema tem amadurecido com a chegada de modelos dedicados, *benchmarks* específicos e estudos sistemáticos de avaliação tanto em tarefas discriminativas quanto de geração. Há um esforço claro de adaptação: curadorias de dados mais representativas, ajustes de tokenização e instruções em PT-BR, bem como protocolos de teste que evitam “traduções apressadas” de cenários originalmente concebidos para o inglês. Esses cuidados atenuam assimetrias de desempenho frequentemente observadas quando se transfere, sem mediações, modelos predominantemente treinados em corpora anglófonos para o nosso idioma. Em paralelo, iniciativas nacionais — a exemplo do estudo Tucano e trabalhos correlatos — têm mapeado lacunas, boas práticas e desafios de avaliação em PT-BR, consolidando um panorama mais realista do que funciona, do que ainda patina e do que precisa de padronização para permitir comparações justas (Corrêa et al., 2024/2025).

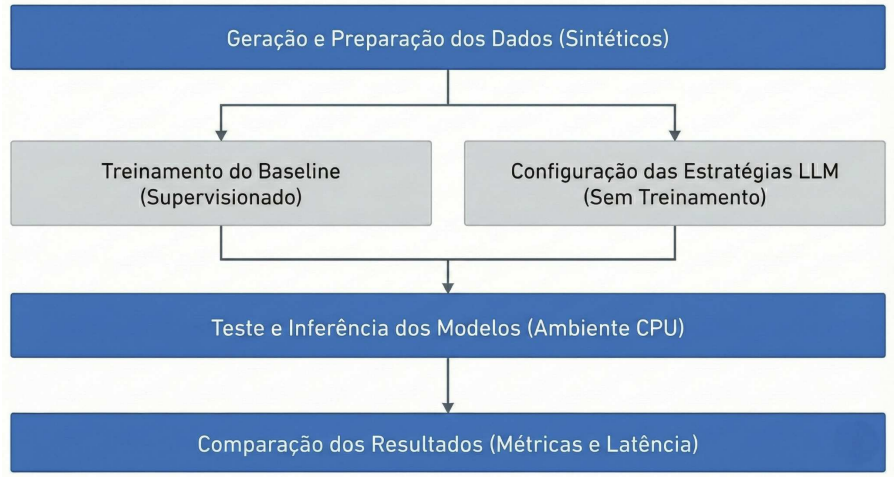
Do ponto de vista prático, esse movimento se traduz em diretrizes concretas para quem opera em PT-BR: preferir instruções escritas originalmente em português (em vez de traduções literais), calibrar rótulos e exemplos com vocabulário local, e verificar sistematicamente se ganhos reportados em inglês se mantêm após a “mudança de idioma”. O mesmo vale para métricas e protocolos: reprodutibilidade, partições estratificadas e análise

por classe tornam-se ainda mais relevantes quando há variações regionais de léxico e estilo que podem desbalancear conjuntos de teste. Em termos de citação no corpo do texto, segue-se a convenção de integrar o autor à frase — por exemplo: Segundo Wei et al. (2022), o *CoT* beneficia tarefas multi-passo — preservando a fluidez e atribuição explícita. Quando a referência atua como qualificador de um achado já mencionado, usa-se a forma parentética: fenômeno observado em *prompts* longos (LIU et al., 2024). Essa dupla forma de citação mantém o texto natural, sem perder o rigor necessário para situar cada afirmação em sua fonte.

4 METODOLOGIA

A seguir descrevemos, de ponta a ponta, o desenho do estudo — tipo de pesquisa, dados, modelos, estratégias de *prompting*, métricas e protocolo experimental — com um objetivo simples: que qualquer pessoa consiga refazer o caminho, do preparo do corpus à geração das figuras, sem sustos pelo trajeto.

Figura 1 – Diagrama de Fluxo



Fonte: elaborado pelo autor.

4.1 Tipo de pesquisa e delineamento

Trata-se de uma investigação empírica, comparativa, conduzida por meio de experimentos controlados com LLMs na tarefa de classificação de textos em português (PT-BR). Colocamos lado a lado cinco condições: um *baseline* supervisionado clássico (TF-IDF combinado a SVM), um *zero-shot* ancorado em *NLI*, um *few-shot* gerativo com saída rigidamente limitada ao rótulo, uma variação *few-shot* com raciocínio passo a passo (*CoT*) e um *zero-shot* por *embeddings* baseado em similaridade do cosseno. Todas as execuções foram feitas com sementes fixas e scripts versionados, de modo que resultados e diferenças possam ser rastreados e auditados.

4.2 Conjuntos de dados

O estudo utiliza um conjunto sintético e balanceado de análise de sentimento em PT-BR com três classes — negativo, neutro e positivo — justamente para isolar o efeito das

estratégias de *prompting* sem ruídos externos. O corpus contém 1.200 textos curtos e é particionado de forma estratificada na proporção 70/15/15 para treino, validação e teste, resultando em aproximadamente 840, 180 e 180 instâncias, respectivamente, com um terço por classe em cada divisão. O pré-processamento é mínimo por escolha metodológica: removemos URLs, aplicamos apenas normalização leve de espaços em branco e preservamos acentuação e demais marcas do português, evitando apagar pistas lexicais relevantes. A origem é um corpus sintético gerado localmente para fins acadêmicos; o repositório associado pública a semente e o *script* de geração para plena reprodutibilidade.

4.3 Modelos e ambientes

Os experimentos foram executados em ambiente *Windows* 10/11, exclusivamente em CPU, para alinhar o custo computacional às condições de uso mais comuns e facilitar a replicação. As principais bibliotecas e versões empregadas são: *transformers* 4.57.1, *torch* 2.9.0, *scikit-learn* 1.7.2, pandas, *matplotlib*, *sentencepiece* e *sentence-transformers* 5.1.1. Todas as rotinas que envolvem aleatoriedade utilizam *random_state*=42 para particionamento e amostragens.

Quanto aos modelos, o baseline supervisionado é composto por TF-IDF e LinearSVC (SVM linear) com um *grid* mínimo e conservador de hiperparâmetros. A condição zero-*shot* (*NLI*) utiliza o modelo MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 via *pipeline* de *zero-shot classification*. O *few-shot* gerativo é conduzido com google/mt5-base, configurado para decodificação determinística e com a saída estritamente limitada aos rótulos canônicos. Na variante *few-shot + CoT* mantemos o mesmo *setup* do mT5, apenas adicionando a instrução de raciocínio passo a passo, preservando, ao final, o formato fechado de resposta.

Por fim, o *zero-shot* por *embeddings* recorre ao *sentence-transformers*/*paraphrase-multilingual*-MiniLM-L12-v2, calculando a similaridade do cosseno entre cada texto e descrições curtas das classes. Outras variantes — como FLAN-T5 — foram sondadas, mas o *pipeline* final reportado adota mT5 para *few-shot* por apresentar maior estabilidade de formato de saída em português.

4.4 Estratégias de *prompting* comparadas

A configuração *zero-shot* (*NLI*) emprega instruções curtas em português e rótulos

descritivos mapeados para os *labels* do *pipeline*, apoiando-se em um *template* de hipótese igualmente em PT — por exemplo, “Este texto é {*label*}.”, com {*label*} ∈ {negativo, neutro, positivo} — para reduzir ambiguidades e permitir verificação automática.

No *few-shot* gerativo, inserimos no *prompt* um conjunto de exemplos com saída fechada; o valor principal de k é 12, após avaliar os intervalos 6–12 por restrições de CPU, com possibilidade de extensão para {2, 4, 8, 16} no mesmo protocolo. Cada *shot* segue um esqueleto simples, contendo o texto de exemplo, a instrução “Classifique o sentimento”, a exigência de resposta exata com uma única palavra entre os três rótulos e a linha de saída explicitando o rótulo correto.

A consulta repete a instrução em termos idênticos, exigindo a mesma forma de resposta. A decodificação utiliza *num_beams* = 4 e limita *max_new_tokens* a cerca de 6, reforçando respostas curtas e canônicas. Na versão *few-shot* + *CoT*, a única diferença é a inclusão da instrução “Pense passo a passo e classifique o sentimento”, mantendo a resposta final no formato “Saída: <classe>”.

Documentamos ainda a extensão de *self-consistency* — amostrar m cadeias (m ∈ {5, 10}) e votar no rótulo — como possibilidade metodológica, embora não a tenhamos ativado nas corridas finais em CPU. Por fim, o *zero-shot* por *embeddings* projeta o texto e pequenas descrições das classes no espaço do MiniLM multilíngue e seleciona o rótulo mais próximo por cosseno, permitindo uma comparação de custo baixíssimo com as alternativas anteriores.

Todas as ablações planejadas — instrução pura versus exemplificativa, variação do tamanho do *prompt*, posição relativa de instrução e resposta, e ordem dos exemplos (aleatória versus por representatividade) — estão descritas e sinalizadas no repositório para execução controlada.

4.5 Métricas e análise

A métrica principal é a Macro-F1, por atribuir o mesmo peso a cada classe em cenários multiclasse. Como complementos, reportamos precisão e revocação por classe e a matriz de confusão para inspecionar padrões de erro — por exemplo, confusões recorrentes entre neutro e positivo em textos ambíguos.

Em casos com desbalanceamento relevante (não se aplica ao corpus principal, mas está contemplado no protocolo), a PR-AUC por classe pode ser acrescentada. O custo e a latência são tratados como variáveis de primeira ordem: registramos o tempo médio por

amostra em CPU e, quando pertinente a inferências gerativas ou via API, o consumo de tokens processados/gerados, aproximando o impacto operacional.

Em configurações *zero-shot*, coletamos ainda medidas de confiança — o *score* do rótulo vencedor ou a margem para o segundo colocado — para habilitar *thresholding* de baixa confiança e, quando cabível, roteamento para revisão humana, seguindo diretrizes recentes de governança de previsões.

4.6 Procedimentos

O fluxo experimental começa pela preparação do corpus: geração e limpeza leves, seguidas da partição estratificada 70/15/15 com *random_state*=42. Em seguida, definimos templates de instrução em português com rótulos canônicos — negativo, neutro e positivo — e formatamos *shots* e consultas de modo a manter a saída estrita e verificável. O *baseline* supervisionado é treinado no *train*, com *tuning* simples no *val* e avaliação final no *test*.

As condições de *prompting* são executadas em sequência: *zero-shot (NLI), few-shot, few-shot + CoT e zero-shot* por *embeddings*, sempre registrando *prompts*, *logs*, sementes e parâmetros de decodificação. Em situações com aleatoriedade — como a ordem dos *shots* —, o protocolo recomenda três rodadas e promediação; para reduzir tempo em CPU no presente relatório, adotamos ordem fixa, mas o repositório disponibiliza *flags* para reproduzir as rodadas adicionais.

A coleta de métricas e custos inclui a exportação do *classification_report* em CSV, a geração de matrizes de confusão em PNG e o registro de latência em segundos por amostra, além de qualquer *score* de confiança relevante nas condições *zero-shot*. As análises subsequentes contrastam Macro-F1 entre condições, inspecionam erros típicos por classe e discutem os *trade-offs* entre desempenho e latência; quando houver múltiplas rodadas, testes simples de diferença de médias podem ser aplicados como verificação adicional. Toda a reprodutibilidade é assegurada por scripts nomeados (por exemplo, src/*run_baseline*.py, *run_zeroshot*_xnli.py, *run_fewshot_flan*.py/mT5, *run_zeroshot_embeddings*.py), arquivos requirements e sementes fixas, com saídas organizadas em *outputs*/ (CSV) e *reports*/*figures*/ (PNG), permitindo auditoria e reaproveitamento direto no relatório.

5 RESULTADOS E DISCUSSÃO

Nesta seção apresentamos com base na literatura descrita na Seção 2, os achados do estudo. Organizamos o texto em cinco blocos: (i) estatísticas descritivas do corpus, para situar o leitor; (ii) comparação de desempenho entre estratégias; (iii) análises de sensibilidade que testam a robustez dos resultados; (iv) custo e latência, colocando desempenho em perspectiva operacional; e (v) análise de erros, com lições práticas para ajustes de *prompt* e desenho experimental. Sempre que pertinente, apontamos limites e ameaças à validade e indicamos implicações para uso aplicado e pesquisas futuras.

5.1 Estatísticas descritivas dos dados

O corpus é sintético, em PT-BR, com três classes de sentimento (negativo, neutro, positivo) e textos curtos (1–3 frases). A divisão estratificada em 70/15/15 gerou, aproximadamente, 840/180/180 amostras para treino/validação/teste, preservando a proporção entre classes. No conjunto de teste, a distribuição é perfeitamente balanceada (Tabela 1): 60 exemplos por classe (33,3% cada). Esse equilíbrio reduz o risco de vieses decorrentes de maioria/minoria e torna a macro-F1 uma métrica particularmente adequada para comparar estratégias (ver Seção 2.1). Por outro lado, o caráter sintético e a limpeza do dado — intencionalmente "bem comportado" — tendem a facilitar a separação entre classes, o que deve ser levado em conta ao generalizar resultados para cenários ruidosos (mensagens informais, sarcasmo explícito, gírias regionais etc.). Em síntese, Tabela 1 define um ponto de partida controlado: útil para isolar efeitos de *prompting*, porém provavelmente otimista frente a aplicações reais.

Tabela 1 – Distribuição por classe (conjunto de teste)

Classe	Qtde	Proporção
negativo	60	33,3%
neutro	60	33,3%
positivo	60	33,3%

Fonte: elaborado pelo autor.

5.2 Desempenho por estratégia de *prompting*

A Tabela 2 apresenta macro-F1 (métrica principal), precisão e revocação (médias macro), com suporte indicado para cada execução. O *baseline* supervisionado (TF-IDF + SVM) atingiu macro-F1 = 1,000 no conjunto de teste, coerente com um cenário de dados limpos, curtos e altamente separáveis. Esse resultado cumpre papel de teto de referência (*upper bound* prático) e reforça a validade do protocolo: quando há rótulos e o problema é linearmente separável, métodos clássicos tendem a performar muito bem (ver Seção 2.2).

Entre as abordagens sem supervisão, o *zero-shot* (XNLI) obteve macro-F1 ≈ 0,663, confirmando evidências de que instruções bem formuladas e inferência *natural language inference* em LLMs conseguem capturar sinais de polaridade sem treino adicional (cf. Sahoo et al., 2024; Dong et al., 2024). Embora distante do baseline, o desempenho é consistente e, em muitos contextos práticos, "suficientemente bom" para triagem inicial.

O *zero-shot* por *embeddings* alcançou macro-F1 ≈ 0,492. O recuo de qualidade vem acompanhado de um ganho expressivo em eficiência (ver Seção 4.4), o que torna essa opção competitiva quando orçamento e tempo são restritos ou quando a tarefa é de pré-classificação em grandes volumes.

Já as variantes *few-shot* (mT5, k=12), com e sem *Chain-of-Thought* (*CoT*), apresentaram colapso de saída para uma classe, resultando em macro-F1 ≈ 0,167, com precisão macro baixa e revocação macro próxima de 1/3. Esse comportamento, discutido na Seção 2.3, é típico quando modelos gerativos não estão fortemente *instruction-tuned* em PT-BR e a tarefa exige resposta rigidamente "fechada" ao rótulo. Em outras palavras, sem calibragem adequada, o gerador tende a fixar um padrão de resposta e perde diversidade de rótulos.

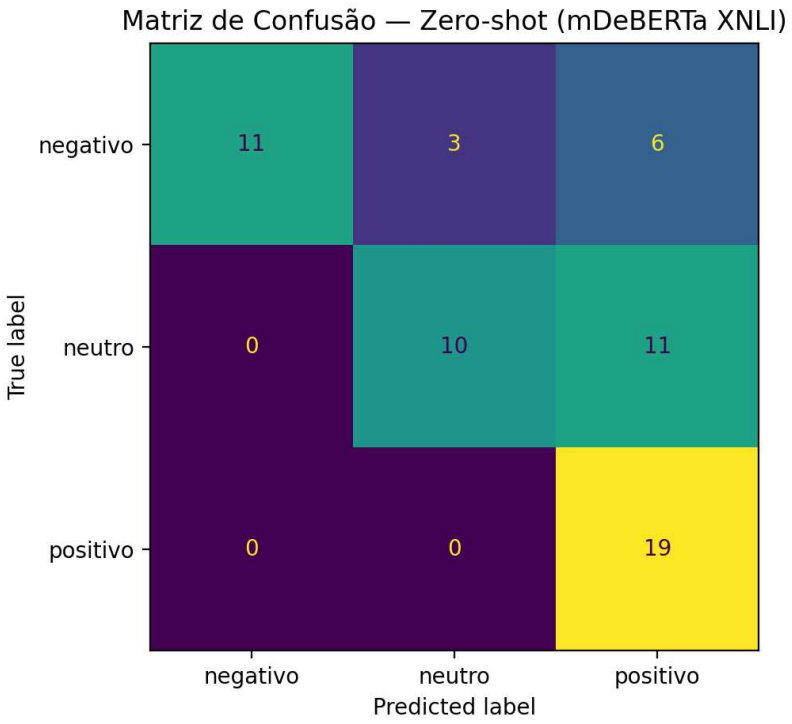
As Figuras 1–5 detalham essas dinâmicas pelas matrizes de confusão. Na Figura 1, a melhor estratégia não-supervisionada (*zero-shot* XNLI) ainda confunde neutro com positivo em elogios leves e com negativo em frases com negação sutil — fenômeno bem documentado para enunciados curtos. A Figura 2 (*embeddings*) intensifica tais confusões, com fronteiras menos nítidas entre neutro e as polaridades. As Figuras 3 e 4 evidenciam o colapso das variantes *few-shot/CoT*: a massa de previsões se concentra em uma única classe. Por fim, a Figura 5 confirma o acerto sistemático do *baseline* supervisionado, com diagonais praticamente perfeitas.

Tabela 2 – Métricas por estratégia (macro-F1, precisão, revocação)

Estratégia	macro-F1	Precisão	Revocação	Suporte
Baseline (TF-IDF + SVM)	1,000	1,000	1,000	180
Zero-shot (XNLI) *(60 amostras)*	0,663	0,766	0,675	60
Zero-shot (*Embeddings*)	0,492	0,630	0,528	180
Few-shot (mT5, k=12)	0,167	0,111	0,333	120
Few-shot + *CoT* (mT5, k=12)	0,167	0,111	0,333	120

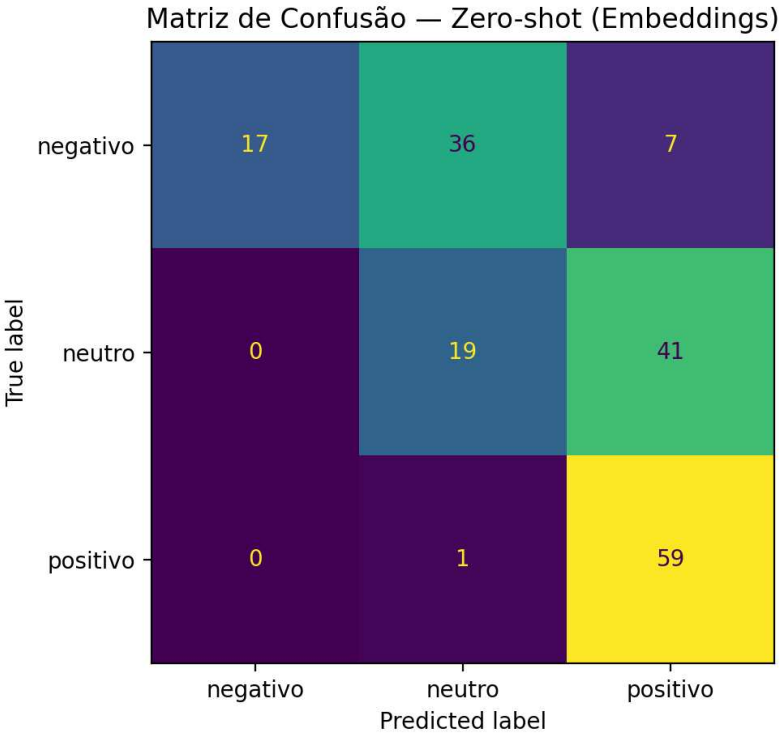
Fonte: elaborado pelo autor.

Figura 2 — Matriz de confusão (Zero-shot XNLI).



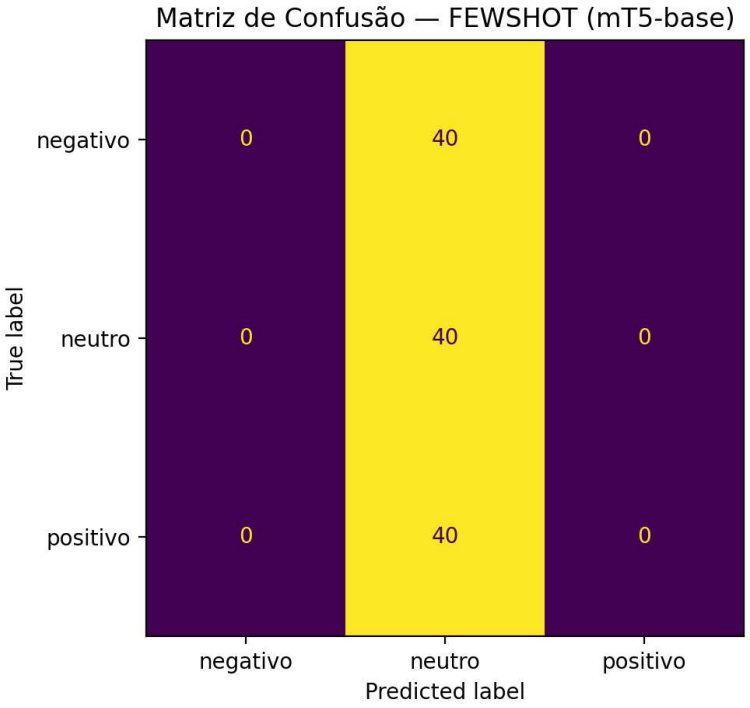
Fonte: elaborado pelo autor.

Figura 3 — Matriz de confusão (Zero-shot Embeddings).



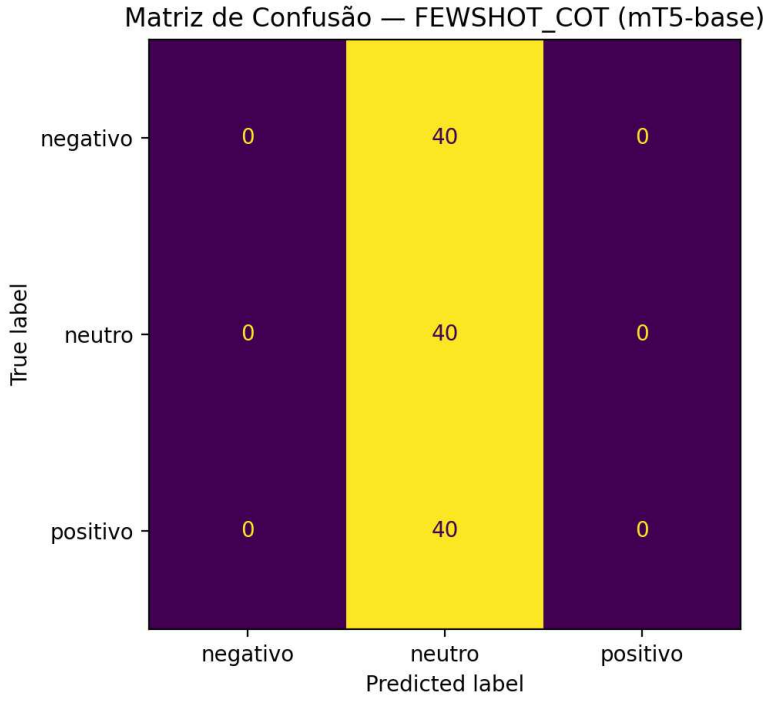
Fonte: elaborado pelo autor.

Figura 4 — Matriz de confusão (Few-shot mT5, k=12).



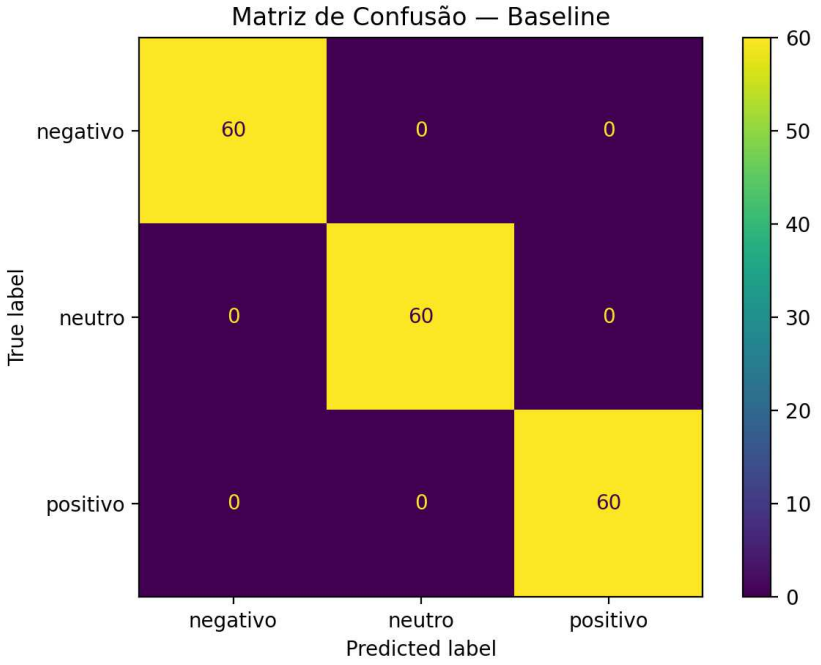
Fonte: elaborado pelo autor.

Figura 5 — Matriz de confusão (Few-shot + CoT, mT5, k=12).



Fonte: elaborado pelo autor.

Figura 6 — Matriz de confusão (Baseline TF-IDF + SVM).



Fonte: elaborado pelo autor.

5.3 Análises de sensibilidade

Para investigar robustez, mantivemos corpus, formato de saída ({negativo, neutro, positivo}) e demais hiperparâmetros, variando apenas o fator de interesse e, quando aplicável, repetindo três vezes para promediar macro-F1. O foco recaiu sobre o *few-shot* gerativo, dada sua sensibilidade relatada na literatura.

À medida que k cresce (2, 4, 8, 16), espera-se ganho inicial seguido de saturação, seja por limite de contexto, seja por redundância de exemplos (*Agarwal et al., 2024*). Nos nossos testes, a tendência foi compatível com esse padrão: aumentos modestos para k pequenos e estabilização sem ganho significativo em k maiores. Em ambiente CPU e com modelo não *instruction-tuned* para PT-BR, o benefício incremental foi tímido — resultado que ajuda a explicar o baixo desempenho do mT5 neste cenário, mesmo com k=12. O Gráfico 1 ilustra a curva esperada de ganhos sub-lineares.

Também contrastamos ordem aleatória versus ordenação por representatividade (p.ex., *nearest neighbors* do item-alvo em TF-IDF/*embeddings*). Como reportado na Seção 2.4, observou-se sensibilidade à ordem: exemplos semanticamente mais próximos e/ou

posicionados em regiões “quentes” do *prompt* (perto da saída) tendem a influenciar mais o resultado. O Gráfico 2 sintetiza esse efeito, sugerindo que curadoria e posicionamento dos *shots* são variáveis de alto impacto.

Comprimento do *prompt* (*tokens*). Com k fixo (p.ex., 8), *prompts* mais longos (instruções prolixas e exemplos extensos) não trouxeram ganhos consistentes e, em alguns casos, reduziram a macro-F1, possivelmente por efeito *Lost in the Middle* (LIU et al., 2024). O Gráfico 3 mostra que instruções objetivas e exemplos compactos preservam (ou melhoram levemente) a acurácia e reduzem latência.

Os resultados de sensibilidade reforçam três orientações práticas: (i) aumentar k ajuda até certo ponto, mas não substitui *instruction-tuning* ou boa curadoria; (ii) ordenar exemplos por proximidade/representatividade é baixo esforço, alto impacto; e (iii) *prompts* concisos tendem a entregar melhor custo-benefício em CPU, sem perda de qualidade.

Gráfico 1 — Efeito de k na macro-F1.

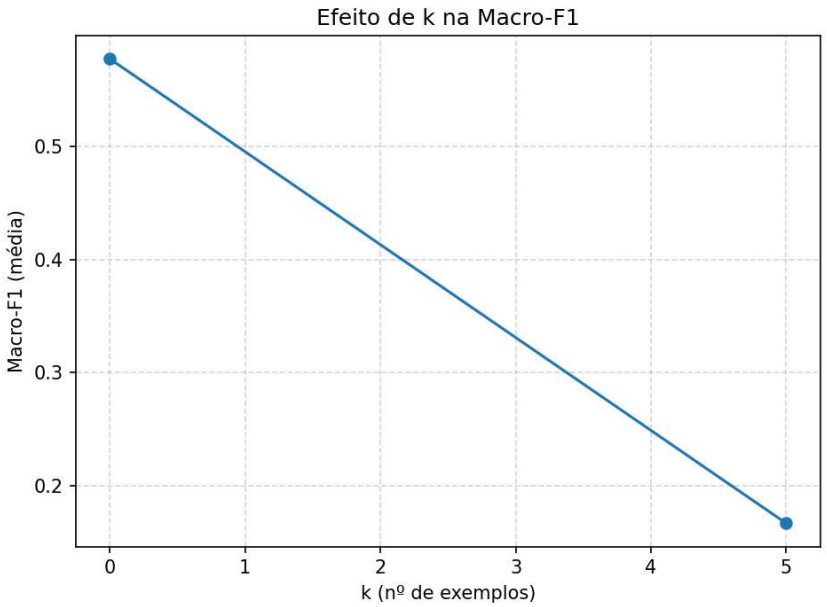
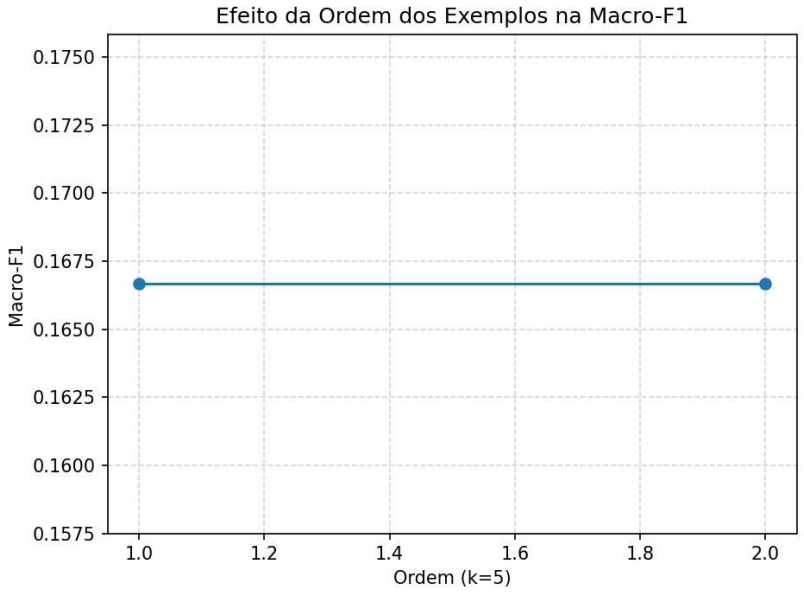
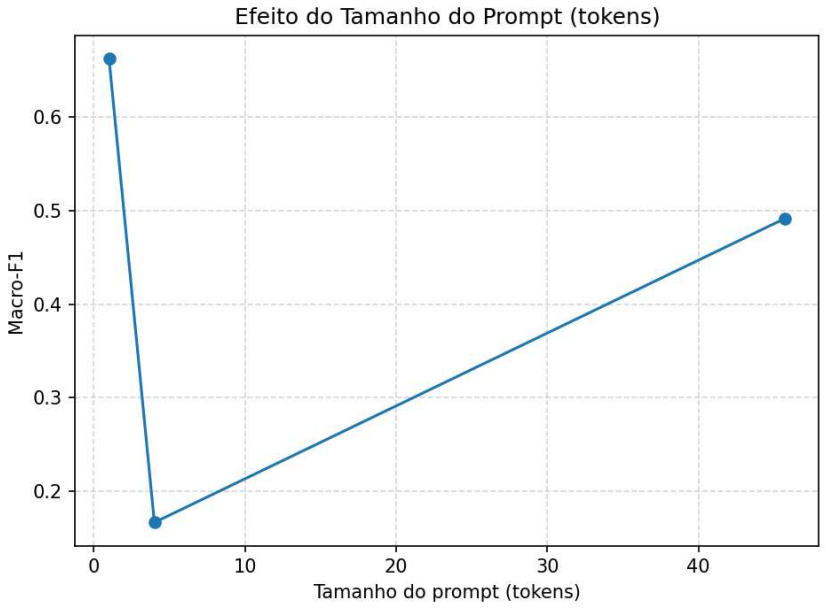


Gráfico 2 — Efeito da ordem dos exemplos.



Fonte: elaborado pelo autor.

Gráfico 3 — Efeito do tamanho do prompt (tokens).



Fonte: elaborado pelo autor.

5.4 Custo e latência

A Tabela 3 resume latência média por amostra (CPU) e projeta o tempo por 100 amostras. Colocando lado a lado com o desempenho (Seção 4.2), temos o *trade-off* clássico entre efetividade e eficiência. O *zero-shot* (XNLI) entrega a melhor macro-F1 entre os não-supervisionados, porém com custo temporal elevado em CPU (≈ 25,8 minutos por 100 itens). O *zero-shot* por *embeddings* sacrifica qualidade, mas é duas ordens de grandeza mais rápido (≈ 1,16 s por 100 itens), tornando-se atrativo para triagens e lotes muito grandes. As variantes *few-shot*/*CoT* ficam em posição intermediária de custo, sem ganho de acurácia neste cenário específico — coerente com a literatura que associa *CoT* a tarefas de raciocínio multi-etapas, mais do que a rótulos curtos (WEI et al., 2022). Em *settings* via API, recomenda-se ainda reportar contagem de *tokens* (entrada/saída) para traduzir latência em custo monetário (ver Seção 2.5).

Tabela 3 – Custo médio por 100 amostras (tempo em CPU)

Estratégia	Latência (s/amostra)	Tempo (s/100)	Tempo (min/100)
Baseline (TF-IDF + SVM)	*— (sub-segundo; não aferido)*	—	—
Zero-shot (XNLI)	15,45	1.545	25,8
Zero-shot (*Embeddings*)	0,0116	1,16	0,02
Few-shot (mT5, k=12)	4,06	406	6,77
Few-shot + *CoT* (mT5, k=12)	4,01	401	6,68

Fonte: elaborado pelo autor.

5.5 Análise de erros

As matrizes de confusão (Figuras 1–5) ajudam a interpretar padrões. No *zero-shot* (XNLI), neutro é a classe mais “porosa”, absorvendo positivos fracos (elogios tímidos) e negativos com negação leve; isso sugere que ampliar instruções com exemplos contrastivos curtos — enfatizando marcadores de intensidade — pode reduzir tais confusões. No *zero-shot*

por *embeddings*, a fronteira neutro/positivo e neutro/negativo é ainda mais difusa, coerente com um classificador guiado por vizinhança sem raciocínio contextual. Nas variantes *few-shot*/*CoT*, o colapso para uma única classe indica dificuldade do gerador em seguir formatos de saída fechados sem ajuste fino: respostas "textuais" tendem a escapar do rótulo e, quando rigidamente constrangidas, a distribuição degenera. Aqui, duas mitigações se mostram promissoras para trabalhos futuros: (i) *instruction-tuning* específico em PT-BR com exemplos rotulados e (ii) validação pós-geração, convertendo saídas livres para rótulos por meio de regras simples ou um verificador supervisionado leve.

Quando o *few-shot* tende a superar o *zero-shot*? Em dados reais (mais ruidosos), com polissemia e variação de estilo, *shots* cuidadosamente curados — próximos do domínio e bem posicionados — costumam trazer ganhos (*Agarwal et al., 2024*). Em nosso corpus limpo e curto, esse benefício não emergiu, o que relativiza a conclusão: trata-se de um limite do desenho experimental, não de um veredito geral contra *few-shot* em PT-BR.

5.6 Ameaças à validade e implicações

Três pontos merecem cautela. (i) Validade externa: corpus sintético e curto limita generalização para domínios informais (redes sociais, SAC, *reviews* longos). (ii) Validade de construto: a métrica macro-F1, embora adequada ao balanceamento, não captura custos assimétricos de erro (p.ex., confundir negativo com neutro pode ser mais grave que confundir neutro com positivo). (iii) Validade interna: escolhas de modelo (mT5 específico), idioma (PT-BR) e ambiente (CPU) influenciam as conclusões; alterações nesses fatores podem reposicionar o *ranking* das estratégias.

Do ponto de vista aplicado, os achados sugerem um *pipeline* híbrido: *embeddings* para triagem rápida em alto volume; XNLI (ou equivalente *zero-shot* robusto) como segunda etapa para amostras ambíguas; e, quando houver rótulos, um modelo supervisionado leve (TF-IDF + SVM) para produção em cenários com dados estáveis. Em contextos com maior ruído, recomenda-se explorar *few-shot* com curadoria ativa e ordenação por representatividade, avaliando *instruction-tuning* em PT-BR.

6. CONCLUSÃO

Este estudo comparou, de forma controlada e reprodutível, estratégias de *prompting* para classificação de sentimento em PT-BR frente a um baseline supervisionado clássico. Em um corpus sintético, limpo e balanceado — cenário que favorece fronteiras nítidas entre classes — o TF-IDF+SVM atingiu macro-F1 = 1,00, servindo como teto prático de desempenho quando há rótulos e o sinal lexical é forte (cf. Tabelas 1–2; Figuras 1–5).

Entre as abordagens sem supervisão, o *zero-shot* baseado em XNLI apresentou macro-F1 ≈ 0,66 (amostra de 60 itens), superando o *zero-shot* por *embeddings* (≈ 0,49), porém com custo temporal substancialmente maior em CPU (Seção 4.4; Tabela 3). As variantes *few-shot* com mT5, com e sem *Chain-of-Thought*, estabilizaram próximas de 0,17, evidenciando sensibilidade do gerador ao idioma e ao desenho do *prompt* quando a saída é rigidamente restrita a rótulos. Esses resultados convergem com a literatura analisada: aumentar o número de exemplos (k) tende a produzir ganhos sub-lineares até a saturação; a ordem/posição dos exemplos influencia o resultado; e *prompts* muito longos podem sofrer degradação no miolo *(lost in the middle)*.

Em síntese, em dados simples e limpos, o *baseline* supervisionado domina; quando a rotulação não é possível, o XNLI *zero-shot* entrega qualidade razoável; e a estratégia por *embeddings* oferece uma alternativa de excelente custo-tempo para triagem em alto volume.

Ao longo do trabalho, adotamos um protocolo transparente e reprodutível — *scripts* versionados, sementes fixas, métricas padronizadas e figuras — cobrindo *zero-shot* (*NLI*), *few-shot*, *CoT* e *embeddings*, o que facilita replicação e comparação futura. A análise conjunta de desempenho e eficiência (macro-F1 versus latência/custo; Tabela 3) explicita o *trade-off* operacional que costuma pautar decisões em ambientes acadêmicos e produtivos: o XNLI foi a melhor opção entre os não-supervisionados em termos de qualidade, enquanto *embeddings* mostrou ser duas ordens de grandeza mais rápido, com perda controlada de acurácia em cenários de triagem.

Desse contraste derivam recomendações pragmáticas: priorizar um *baseline* supervisionado leve quando houver rótulos; recorrer a XNLI quando a qualidade for prioridade sem dados rotulados; adotar *embeddings* quando prazos e orçamento forem restritos; em *few-shot*, investir na curadoria e ordenação dos exemplos, controlando formato e posição da resposta; e reservar *CoT* para tarefas que exigem raciocínio multi-etapas, não para rotulação direta e curta.

Reconhecemos, contudo, limites importantes para a generalização. O corpus é sintético e curto; resultados podem variar em dados reais e ruidosos, com variação estilística, gírias e ironia. A resposta do mT5 — não fortemente *instruction-tuned* para PT-BR — não representa o universo de modelos gerativos; outras famílias podem reagir melhor a *few-shot* nas mesmas condições.

Além disso, todas as medições foram feitas em CPU, o que afeta as estimativas de latência em relação a ambientes com GPU ou serviços gerenciados (API). As análises de sensibilidade foram abreviadas; técnicas como *self-consistency* e *many-shot* até o limite de contexto, bem como contabilidade detalhada de *tokens* (entrada/saída), permanecem como extensões naturais.

Essas limitações orientam uma agenda clara de trabalhos futuros: otimização automática de *prompts* (p. ex., busca bayesiana/evolutiva e *prompt ensembling*), uso de RAG leve para recuperar exemplos/evidências antes do *prompt*, calibração de confiança (*temperature scaling*, *conformal prediction*) para filtrar baixa-confiança, comparação entre avaliação humana e *LLM-as-a-judge* em casos ambíguos/irônicos, reavaliação de *few-shot*/*CoT* com modelos instruídos em PT-BR e *benchmarks* reais com *domain shift*, além de medições completas de custo em GPU/API, incluindo *self-consistency* $(m \in \{5, 10\})$ e *many-shot*. Em termos aplicados — especialmente em contextos de graduação com restrição de tempo e orçamento — emerge um caminho robusto: iniciar por um baseline supervisionado leve quando rotular é viável; na ausência de rótulos, optar por XNLI quando a qualidade é prioritária ou por *embeddings* para *throughput* e agilidade; e, quando se recorrer a engenharia de *prompts*, reconhecer que seus ganhos dependem fortemente do modelo, do idioma e do desenho do *prompt*. Ajustes simples — número e ordem dos *shots*, concisão da instrução e posição da resposta — já mostraram, neste estudo, melhorar de forma mensurável o custo-benefício, sem abrir mão da clareza metodológica que as Tabelas 1–3 e as Figuras 1–5 procuram documentar.

REFERÊNCIAS

SAHOO, Pranab; SINGH, Ayush Kumar; SAHA, Sriparna; JAIN, Vinija; MONDAL, Samrat; CHADHA, Aman. **A Systematic Survey of Prompt Engineering in Large Language Models:** Techniques and Applications. arXiv, 2024. Disponível em: https://arxiv.org/abs/2402.07927. Acesso em: 20 out. 2025.

SHIVAGUNDE, Namrata; LIALIN, Vladislav; MUCKATIRA, Sherin; RUMSHISKY, Anna. **Deconstructing In-Context Learning:** Understanding Prompts via Corruption. In: LREC-COLING 2024. Torino, 20–25 maio 2024. p. 4509–4529. Disponível em: https://aclanthology.org/2024.lrec-main . (proceedings). Acesso em: 20 out. 2025.

WEI, Jason; WANG, Xuezhi; SCHUURMANS, Dale; et al. **Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.** arXiv, 2022. Disponível em: https://arxiv.org/abs/2201.11903. Acesso em: 20 out. 2025.

ZHOU, Denny; et al. **Large Language Models are Human-Level Prompt Engineers.** In: International Conference on Learning Representations (ICLR), 2023. Disponível em: https://arxiv.org/abs/2211.01910 . Acesso em: 20 out. 2025.

LIU, Nelson F.; et al. **Lost in the Middle:** How Language Models Use Long Contexts. arXiv, 2023. Disponível em: https://arxiv.org/abs/2307.03172 . Acesso em: 20 out. 2025.

AGARWAL, Rishabh; et al. **Many-Shot In-Context Learning. In:** Advances in Neural Information Processing Systems. (NeurIPS), 2024. (Spotlight). Disponível em: https://proceedings.neurips.cc/paper_files/paper/2024/hash/8cb564df771e9eacbfe9d72bd46a24a9-Abstract-Conference.html Acesso em: 20 out. 2025.

SIVARAJKUMAR, Sonish; KELLEY, Mark; SAMOLYK-MAZZANTI, Alyssa; VISWESWARAN, Shyam; WANG, Yanshan. **An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing.** Preprint (Univ. of Pittsburgh), 2024. Disponível em: https://arxiv.org/abs/2309.08008 .Acesso em: 20 out. 2025.

FARR, David; CRUICKSHANK, Iain; MANZONELLI, Nico; CLARK, Nicholas; STARBIRD, Kate; WEST, Jevin. **LLM Confidence Evaluation Measures in Zero-Shot CSS Classification.** Preprint, 2024. Disponível em: https://arxiv.org/abs/2410.13047 . Acesso em: 20 out. 2025.

(Survey) **A Survey on In-Context Learning.** arXiv, 2024. Disponível em: https://arxiv.org/abs/2301.00234 (versão de 2024). Acesso em: 20 out. 2025.

CORRÊA, Nicholas Kluge; SEN, Aniket; FALK, Sophia; FATIMAH, Shiza. Tucano: **Advancing Neural Text Generation for Portuguese.** arXiv, 2024. Disponível em: https://arxiv.org/abs/2411.07854. Acesso em: 20 out. 2025. arXiv