Revista Eletrônica de Iniciação Científica em Computação, 2026, 24:1
ISSN: 3085-8461 • doi: 10.5753/reic.2026.8461

Artigo de Pesquisa/Research Paper

Um Estudo Comparativo de Estratégias de Seleção de Exemplos para *In-Context Learning* Aplicado à Classificação Automática de Texto com LLMs

A Comparative Study of Example Selection Strategies for In-Context Learning Applied to Automatic Text Classification with LLMs

Gabriel Prenassi [Universidade Federal de São João del-Rei | *prenassigabriel@aluno.ufsj.edu.br*]
Guilherme Fonseca [Universidade Federal de Minas Gerais | *guilhermefonseca@dcc.ufmg.br*]
Marcos André Gonçalves [Universidade Federal de Minas Gerais | *mgoncalv@dcc.ufmg.br*]
Leonardo Rocha [Universidade Federal de São João del-Rei | *lcrocha@ufsj.edu.br*]

Departamento de Ciência da Computação, Universidade Federal de São João del-Rei, BR-494, s/n, Campus CTan, São João del-Rei, MG, 36301-360, Brasil.

Resumo. A Classificação Automática de Texto com LLMs pode ser realizada via zero-shot (ZS) (baixo custo e menor desempenho) ou por fine-tuning (FT) (maior custo e maior desempenho). Este trabalho investiga o in-context learning (ICL) como alternativa intermediária, analisando o trade-off entre efetividade e eficiência. Avaliamos estratégias de seleção de exemplos, comparando abordagem aleatória e métodos baseados em representações vetoriais (TF-IDF, RoBERTa, SBERT e LLM2Vec), além da variação do número de exemplos nos prompts. Os resultados indicam que boas estratégias de seleção tornam o ICL promissor, superando o ZS em efetividade com custo inferior ao FT. Ainda assim, o melhor custo-benefício permanece com o FT de SLMs como o RoBERTa.

Abstract. Text Classification with LLMs can be performed via zero-shot (ZS) (lower cost and lower performance) or through fine-tuning (FT) (higher cost and higher performance). This work investigates in-context learning (ICL) as an intermediate alternative, analyzing the trade-off between effectiveness and efficiency. We evaluate example selection strategies, comparing a random approach with methods based on vector representations (TF-IDF, RoBERTa, SBERT, and LLM2Vec), as well as varying the number of examples included in the prompts. The results indicate that well-designed selection strategies make ICL promising, outperforming ZS in effectiveness while incurring lower cost than FT. Nevertheless, the best cost-benefit ratio remains with fine-tuning SLMs such as RoBERTa.

Palavras-chave: Classificação Automática de Texto, Grandes Modelos de Linguagem, *In-Context Learning*, Seleção de Exemplos, Representações Vetoriais

Keywords: Text Classification, Large Language Models, In-Context Learning, Example Selection, Vector Representations

Recebido/Received: 14 June 2026 • **Aceito/Accepted:** 16 June 2026 • **Publicado/Published:** 10 July 2026

1 Introdução

A Classificação Automática de Texto (CAT) constitui uma tarefa central em Processamento de Linguagem Natural, com aplicações em diferentes cenários, como categorização de documentos e detecção de discurso de ódio. O interesse crescente nessas aplicações impulsiona o desenvolvimento de métodos cada vez mais sofisticados para a tarefa [Cunha *et al*., 2025b]. Nos últimos anos, avanços expressivos passaram a ser observados, especialmente com a adoção de arquiteturas neurais baseadas em *Transformers*. Inicialmente, destacaram-se modelos como BERT e RoBERTa [Devlin *et al*., 2019; Liu *et al*., 2019], aqui referidos como *Small Language Models* (SLMs). Posteriormente, os *Large Language Models* (LLMs), como GPT e Llama [OpenAI *et al*., 2024; Grattafiori *et al*., 2024], ampliaram a capacidade de generalização e adaptação a diferentes tarefas. Atualmente, os LLMs representam o estado da arte em CAT [Cunha *et al*., 2025b].

Diante desse cenário, diferentes estratégias têm sido exploradas para empregar LLMs em tarefas de CAT, conforme ilustrado na Figura 1. Uma possibilidade é o uso em *zero-shot* (Figura 1a), no qual o modelo pré-treinado é utilizado por meio de *prompts*, sem adaptação adicional ao domínio da tarefa, resultando em menor custo computacional, porém com possíveis limitações de desempenho [Lu *et al*., 2024; Edwards and Camacho-Collados, 2024; Chandra *et al*., 2025]. Outra estratégia consiste no *fine-tuning* supervisionado (Figura 1c), em que os parâmetros do modelo são ajustados com dados rotulados, aumentando o grau de especialização ao custo de maior demanda computacional [Cunha *et al*., 2025b]. Entre essas abordagens, o *in-context learning* (ICL) (Figura 1b) incorpora exemplos diretamente no *prompt*, preservando os pesos originais do modelo e buscando conciliar custo e desempenho [Lu *et al*., 2024; Edwards and Camacho-Collados, 2024; Xu *et al*., 2024; Chandra *et al*., 2025]. Essa última abordagem constitui o foco principal deste estudo, a partir do qual se estruturam as seguintes perguntas de pesquisa:

PP1: *Qual a relação entre efetividade e eficiência dos paradigmas zero-shot e fine-tuning em CAT com LLMs?*
PP2: *Em que medida o in-context learning, variando o número de exemplos no prompt, pode otimizar esse trade-off?*
PP3: *Como diferentes estratégias de representação impactam o desempenho do in-context learning?*

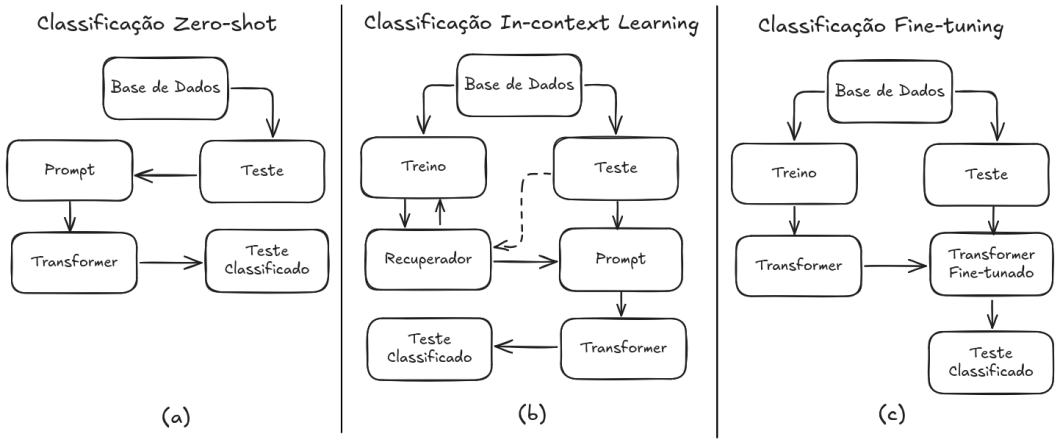

Figura 1. Comparação conceitual de abordagens de CAT: *zero-shot*, *in-context learning* e *fine-tuning*. Em (a), o texto é classificado diretamente pelo LLM a partir de um *prompt*, sem ajuste de pesos e sem exemplos rotulados do conjunto de treinamento. Em (b), exemplos rotulados são incorporados ao *prompt* antes da classificação. A seta pontilhada indica que, em alguns métodos de ICL, são recuperados exemplos semelhantes ao texto a ser classificado. Em (c), primeiro ocorre uma etapa de treinamento, na qual os pesos do modelo são ajustados com base em dados rotulados. Em seguida, o modelo já ajustado é utilizado para classificar novos textos.

Para respondê-las, realizamos uma comparação entre: (i) *zero-shot*, utilizando Llama 3.1 [Grattafiori *et al*., 2024]; (ii) *fine-tuning*, com Llama 3.1 e RoBERTa [Liu *et al*., 2019], apontados na literatura como referências em CAT [Cunha *et al*., 2023; Fonseca *et al*., 2025; Cunha *et al*., 2025b,a]; e (iii) ICL, no qual exploramos diferentes estratégias de seleção de exemplos, desde amostragem aleatória até abordagens baseadas em representações vetoriais, como TF-IDF [Luhn, 1957], RoBERTa, SBERT [Chandra *et al*., 2025] e LLM2Vec [BehnamGhader *et al*., 2024], além de variações na quantidade de exemplos por *prompt*. **Até onde se tem conhecimento, uma análise sistemática comparando múltiplas estratégias de ICL sob essas diferentes configurações ainda é limitada na literatura, constituindo a principal contribuição deste trabalho.**

Nossos resultados corroboram achados recentes da literatura [Cunha *et al*., 2025b], indicando que SLMs com *fine-tuning* oferecem a melhor relação entre efetividade e eficiência em CAT (**PP1**). Observamos também que o ICL se configura como uma alternativa promissora entre *zero-shot* e *fine-tuning* (**PP2**), apresentando, de modo geral, desempenho superior ao *zero-shot* à medida que aumenta o número de exemplos no *prompt*, embora tal comportamento dependa das características dos dados e das restrições computacionais. Adicionalmente, estratégias de seleção baseadas em representações vetoriais mostram-se capazes de gerar ganhos consistentes com custo computacional moderado (**PP3**). Por fim, os resultados evidenciam oportunidades de aprimoramento das abordagens de ICL, especialmente por meio do desenvolvimento de novos métodos de representação e seleção de exemplos. Este trabalho de iniciação científica resultou na publicação de um artigo no SBBD 2025 (A4) [Prenassi *et al*., 2025], tendo recebido menção honrosa como melhor artigo curto do evento.

2 Trabalhos Relacionados

Na última década, a CAT passou por avanços expressivos, impulsionados principalmente pelo surgimento de SLMs e LLMs [Cunha *et al*., 2025b]. Nesse contexto, o ICL tem sido investigado como alternativa intermediária entre *zero-shot* e *fine-tuning*, direcionando esforços de pesquisa para a definição de estratégias de construção de *prompts*, com o objetivo de equilibrar custo computacional e desempenho.

Nesse contexto, a literatura pode ser organizada segundo o critério adotado para seleção dos exemplos inseridos no *prompt*. Quatro trabalhos utilizam seleção aleatória, ainda que com propostas distintas. Em [Edwards and Camacho-Collados, 2024], insere-se um exemplo por classe com o objetivo de comparar *prompting* e *fine-tuning*. Já [Xu *et al*., 2024] combina LLMs com SLMs ajustados localmente que atuam como *plug-ins*, mantendo a escolha aleatória e variando a quantidade de exemplos conforme o conjunto de dados. Em [Lu *et al*., 2024], propõe-se um *framework* em duas etapas, envolvendo autorredução do espaço de rótulos e comparações contrastivas par a par entre os rótulos restantes, empregando 3 ou 5 exemplos também selecionados aleatoriamente, a depender do LLM. Por sua vez, [Kumar and Talukdar, 2021] investiga o impacto da ordem dos exemplos no *prompt*, mantendo a seleção aleatória e sem definir de forma sistemática a quantidade utilizada.

Em contraste, dois estudos adotam critérios baseados na similaridade. O trabalho de [Liu *et al*., 2022] utiliza representações geradas por RoBERTa *fine-tuned* para recuperar instâncias semanticamente próximas ao texto a ser classificado, com variação na quantidade de exemplos conforme o cenário. Já [Chandra *et al*., 2025] propõe a predição da quantidade ideal de exemplos por meio de um classificador multirótulo que mapeia características das instâncias de teste, como *embeddings* e distribuições de rótulos de vizinhos, realizando a seleção por similaridade com SBERT.

Além do critério de seleção, destacam-se dois aspectos complementares: a ordem e a quantidade de exemplos no *prompt*. Enquanto [Kumar and Talukdar, 2021] aponta que a ordenação pode impactar a generalização, apenas [Chandra *et al*., 2025] aborda explicitamente a definição sistemática da quantidade de exemplos. Nos demais casos, essa quantidade é fixada ou variada sem um critério estruturado.

Assim, diferentemente dessas abordagens, este trabalho investiga, de forma conjunta, dois aspectos frequentemente tratados separadamente: (i) a variação sistemática na quantidade de exemplos no *prompt*; e (ii) diferentes estratégias

de seleção, em sua maioria baseadas em similaridade. Para isso, além de representações já exploradas na literatura, como RoBERTa e SBERT, consideram-se abordagens clássicas, como TF-IDF [Luhn, 1957], e técnicas mais recentes, como LLM2Vec [BehnamGhader *et al*., 2024], que exploram a capacidade semântica dos LLMs para a geração de *embeddings*. Como linha de base, também é avaliada a seleção aleatória de exemplos.

3 Metodologia Experimental

3.1 Bases de Dados

Os experimentos foram conduzidos utilizando quatro bases de dados, descritas na Tabela 1. Essas coleções diferem entre si quanto ao número de documentos, à dimensionalidade (tamanho do vocabulário), ao número de classes, à densidade média de palavras por documento e à distribuição das classes.

Base	# Docs.	Dim.	# Classes	Dens.	Distrib.
TREC	5.952	3.032	6	10	Desbalanceada
Twitter	6.997	8.135	6	28	Desbalanceada
SST-1	11.855	9.015	5	19	Balanceada
ACM	24.897	48.867	11	65	Desbalanceada

Tabela 1. Bases de dados utilizadas nos experimentos. As abreviações indicam número de documentos (# Docs.), dimensionalidade (Dim.), densidade média (Dens.) e distribuição das classes (Distrib.).

Para complementar essa caracterização quantitativa, também consideramos a origem e o domínio textual das bases. Nesse sentido, a TREC reúne perguntas curtas de conhecimento geral, classificadas conforme o tipo de resposta esperada. A base Twitter, por sua vez, contém postagens de rede social, enquanto a SST-1 é derivada de avaliações com rótulos de sentimento. Já a ACM representa um domínio mais específico, composta por artigos acadêmicos da área de computação. Dessa forma, as coleções não convergem para um único assunto ou tipo textual, abrangendo cenários com diferentes níveis de formalidade, densidade e homogeneidade temática.

3.2 Estratégia de Construção dos Prompts

Avaliamos o desempenho da abordagem *zero-shot* utilizando o *prompt* apresentado na Figura 2. Nesse cenário, incluímos apenas um exemplo genérico, sem relação com os dados de treinamento, com o objetivo de indicar o formato esperado de resposta (A, B, C etc. correspondem aos rótulos), mantendo-o fixo para todos os documentos classificados.

Por outro lado, na abordagem ICL, utilizamos a mesma estrutura de *prompt*, com inserção progressiva de 10 a 100 exemplos, em intervalos de 10. Quanto à forma de utilização dos exemplos no *prompt*, consideramos dois procedimentos. Na abordagem aleatória, definimos o conjunto de exemplos uma única vez a partir do conjunto de treinamento, reutilizando-o para todos os documentos classificados. Já nas abordagens baseadas em representações vetoriais (TF-IDF ou *embeddings* gerados pelos modelos), definimos os exemplos individualmente para cada instância de teste. Para cada documento, aplicamos KNN sobre o conjunto de treinamento, utilizando a similaridade cosseno, a fim de recuperar os exemplos semanticamente mais próximos, ordenados por similaridade decrescente. Assim, nesses casos, cada documento foi associado a um *prompt* específico.

Prompt

Classify the topic of the text exclusively among the references.
Input: {example text 1}
A. {class name 1}
B. {class name 2}
⋮
{letter corresponding to the class name of example text 1}
...
Input: {example text K}
A. {class name 1}
B. {class name 2}
⋮
{letter corresponding to the class name of example text K}
Input: {text to be classified}

Figura 2. Para *zero-shot*, utilizou-se apenas um exemplo que não pertence ao treino. Já para *in-context learning*, foram usados até K exemplos do treino.

3.3 Modelos e Configurações Experimentais

Adotamos o Llama 3.1 como LLM por se tratar de um modelo aberto e por apresentar desempenho de destaque em tarefas de CAT [Cunha *et al*., 2025b]. Nos experimentos de *fine-tuning*, utilizamos a versão Llama-3.1-8B, enquanto nos cenários de *zero-shot* e ICL empregamos a variante Llama-3.1-8B-Instruct, otimizada para *prompting*. Em ambos os cenários com Llama, aplicamos quantização em 4 *bits*, reduzindo o consumo de memória da GPU sem impacto relevante no desempenho [Hu *et al*., 2021]. Como SLM de referência, utilizamos o RoBERTa, amplamente reconhecido na literatura como um forte *baseline* para CAT [Cunha *et al*., 2023, 2025b]. Nos experimentos de *fine-tuning* de Llama e RoBERTa, replicamos os hiperparâmetros reportados em [Cunha *et al*., 2025b], os quais são apresentados na Tabela 2, junto com os parâmetros utilizados no processo de geração de respostas nos cenários de *zero-shot* e ICL.

Para a construção das representações vetoriais empregadas nas estratégias de ICL baseadas em similaridade, consideramos quatro modelos: RoBERTa, RoBERTa *fine-tuned*, SBERT (*all-MiniLM-L6-v2*) [Reimers and Gurevych, 2019] e LLM2Vec (LLM2Vec-Meta-Llama-31-8B-Instruct-mntp) [BehnamGhader *et al*., 2024]. As respectivas configurações adotadas para cada modelo também estão detalhadas na Tabela 2.

Modelo	Aplicação	Configuração
Llama-3.1-8B	Fine-tuning	lr: 2×10^{-4}; epochs: 4; batch_size: 4; max_len: 256; quant.: 4-bit
Llama-3.1-8B-Instruct	*Zero-shot* / ICL	max_new_tokens: 1; do_sample: True; temp.: 0.1; top_p: 0.9; quant.: 4-bit
RoBERTa	Fine-tuning	lr: 5×10^{-5}; max_epochs: 20; patience: 5; grid search: max_len $\in \{128, 256\}$; batch_size $\in \{16, 32\}$
RoBERTa	Representação	grid search: max_len $\in \{128, 256\}$; batch_size $\in \{16, 32\}$
RoBERTa *fine-tuned*	Representação	mesmos hiperparâmetros do *fine-tuning*
SBERT	Representação	configuração padrão do modelo
LLM2Vec	Representação	pooling: mean; max_length: 512

Tabela 2. Configurações de treinamento, inferência e representação.

3.4 Métrica de Avaliação

Para fins de comparação, os resultados das abordagens *zero-shot* e ICL foram confrontados com o desempenho dos modelos RoBERTa e Llama-3.1-8B após *fine-tuning*.

A avaliação considerou duas dimensões complementares: efetividade e eficiência. A efetividade foi mensurada por meio da Macro-F1, métrica recomendada para cenários com classes desbalanceadas, enquanto a eficiência foi estimada a partir do tempo total de execução (em segundos), considerando as particularidades de cada abordagem.

Especificamente, no *zero-shot*, contabilizamos apenas o tempo de classificação, uma vez que não há etapa de treinamento. No ICL, quando empregada alguma forma de representação, o tempo inclui a geração das representações dos conjuntos de treino e teste, além da etapa de classificação. No caso da estratégia baseada em representações do RoBERTa *fine-tuned*, incorporamos também o tempo necessário para o ajuste dos pesos do modelo. Na seleção aleatória, consideramos apenas os tempos de seleção e de classificação. Já na abordagem de *fine-tuning*, tanto para LLMs quanto para SLMs, o tempo reportado corresponde à soma da etapa prévia de ajuste dos pesos e da classificação posterior com o modelo já ajustado. Essa escolha busca medir o custo total de aplicação da abordagem no cenário experimental avaliado, e não apenas o tempo de classificação após o ajuste.

Os experimentos foram conduzidos com validação cruzada estratificada, utilizando 10 *folds*, em uma instância AWS g5.2xlarge (8 vCPUs, 32 GiB de RAM e uma GPU NVIDIA A10G com 24 GiB de memória). As diferenças observadas nos resultados foram avaliadas por meio de teste-t pareado, com nível de confiança de 95%, aplicando-se a correção de Bonferroni [Hochberg, 1988]. Assim, os resultados de Macro-F1 são apresentados como média acompanhada do intervalo de confiança de 95%, enquanto os tempos de execução correspondem apenas à média entre os *folds*.

4 Resultados

Os resultados[1] das avaliações descritas na seção anterior estão apresentados nas Figuras 3 e 4. A primeira apresenta a efetividade, mensurada por Macro-F1 (eixo y), enquanto a segunda reúne os resultados de eficiência, medidos pelo tempo de execução (eixo y). Em ambos os casos, as análises consideram a variação da quantidade de exemplos do conjunto de treinamento utilizados nos *prompts* (eixo x), sendo $x = 0$ correspondente ao cenário *zero-shot*. As linhas azul e laranja representam, respectivamente, os modelos Llama-3.1-8B e RoBERTa após *fine-tuning*. Como esses modelos foram treinados com todo o conjunto de treinamento e não utilizam estratégias de *prompting* durante a inferência, seus valores de Macro-F1 e tempo permanecem constantes em relação à quantidade de exemplos utilizada.

4.1 *Zero-Shot versus Fine-Tuning*

A Figura 3 evidencia diferenças de efetividade entre a abordagem *zero-shot* e aquelas baseadas exclusivamente em *fine-tuning*, com os modelos ajustados apresentando desempenho superior. Mais especificamente, Llama e RoBERTa exibem resultados bastante próximos nas bases Twitter (78.6±1.6 vs. 78.4±1.8) e TREC (96.1±0.8 vs. 95.5±0.5). Já nas bases SST-1 e ACM, o Llama apresenta vantagem mais expressiva (58.7±1.0 vs. 53.8±1.3 e 77.8±0.9 vs. 70.3±1.4, respectivamente), indicando maior capacidade de generalização nesses cenários. Entretanto, ao analisar os resultados de eficiência (Figura 4), observamos que o custo computacional do *fine-tuning* do Llama é substancialmente superior tanto ao cenário *zero-shot* quanto ao RoBERTa *fine-tuned*, em todas as bases avaliadas. Na base ACM, por exemplo, o tempo médio foi de 380,98 segundos para o *zero-shot*, 1.043,31 para o RoBERTa *fine-tuned* e 13.470,88 para o Llama *fine-tuned*.

Em conjunto, esses resultados respondem à PP1 ao evidenciar que o *fine-tuning* proporciona ganhos consistentes de efetividade em relação ao *zero-shot*, porém, à custa de maior demanda computacional, especialmente no caso do Llama. Assim, embora o LLM apresente vantagem em algumas bases, os resultados indicam que SLMs como o RoBERTa mantêm relação mais favorável entre desempenho e custo computacional, alinhando-se a evidências previamente reportadas [Cunha *et al*., 2025b].

4.2 *In-Context Learning* entre *Zero-Shot* e *Fine-Tuning*

Com relação à PP2, analisamos em que medida o ICL, ao variar a quantidade de exemplos no *prompt*, é capaz de otimizar o *trade-off* entre efetividade e eficiência. Para isso, avaliamos seis estratégias distintas, variando progressivamente o número de exemplos inseridos no *prompt*. De modo geral, observa-se que o aumento do número de exemplos tende a melhorar a efetividade em relação ao *zero-shot*. Contudo, os resultados indicam que não existe um número ótimo universal de exemplos, uma vez que o desempenho depende tanto das características da coleção quanto da estratégia de seleção adotada (Figura 3).

Em termos de eficiência (Figura 4), embora o *overhead* associado à seleção e inserção de exemplos aumente o tempo de execução em relação ao *zero-shot*, esse custo permanece substancialmente inferior ao requerido pelo *fine-tuning* de LLMs. Dessa forma, o ICL posiciona-se como alternativa intermediária entre esses dois paradigmas. Contudo, observa-se que, na coleção ACM, configurações com mais de 40 exemplos tornaram-se inviáveis devido a restrições de memória da GPU, associadas à elevada densidade e dimensionalidade do conjunto. Esse resultado evidencia limitações práticas do método em determinados cenários e acende um alerta quanto à sua usabilidade.

4.3 Impacto das Representações no Desempenho do *In-Context Learning*

Com relação à PP3, analisamos como diferentes estratégias de representação impactam o desempenho do ICL. Os resultados apresentados na Figura 3 evidenciam que a escolha da representação exerce influência direta e significativa na efetividade alcançada. Em todas as quatro coleções avaliadas, as abordagens baseadas em representações vetoriais superaram consistentemente a seleção aleatória, especialmente à medida que aumentava o número de exemplos nos *prompts*. Assim, embora a seleção randômica seja frequentemente adotada na literatura (Seção 2), neste estudo ela apresentou, de forma sistemática, os menores níveis de efetividade, sem redução relevante no tempo de execução (Figura 4). Esses achados indicam que a seleção específica por instância, orientada por representações vetoriais e critérios de similaridade, é

[1]Os resultados em formato tabular estão disponíveis em: https://github.com/gabrielprenassi/icl_cat_tradeoff

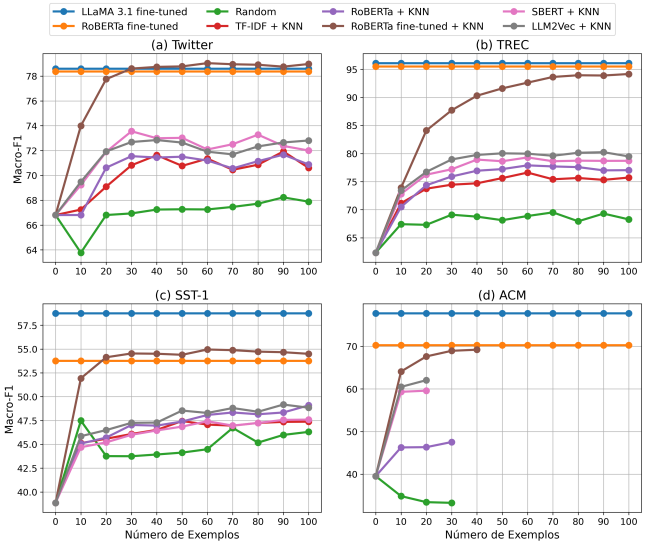

Figura 3. Comparação da Efetividade (Macro-F1) entre abordagens de CAT.

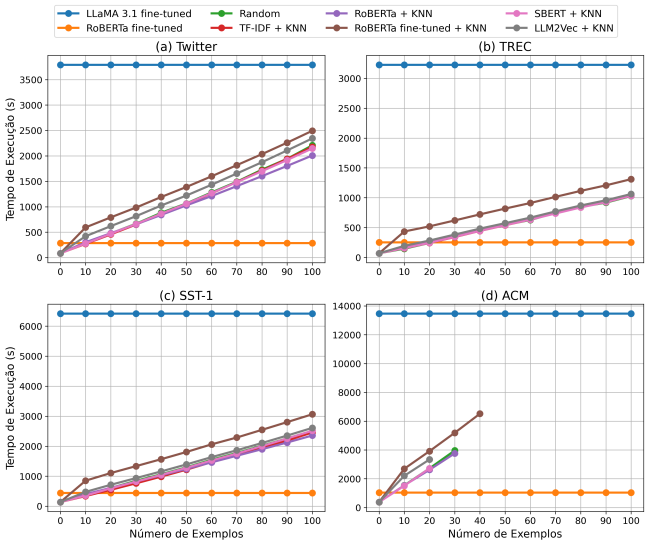

Figura 4. Comparação da Eficiência (Tempo) entre abordagens de CAT.

determinante para o desempenho do ICL.

Entre as estratégias vetoriais, a combinação *RoBERTa fine-tuned + KNN* destacou-se por se aproximar, em diversos cenários, do desempenho de modelos totalmente ajustados, mesmo com poucos exemplos nos *prompts*. Na base Twitter, por exemplo, com 30 exemplos, essa estratégia alcança 78.6±1.1, desempenho estatisticamente equivalente e próximo ao obtido pelo *LLaMA 3.1 fine-tuned* (78.6±1.6). Em termos de tempo, contudo, observa-se uma diferença relevante: 983,56 s para *RoBERTa fine-tuned + KNN*, 285,64 s para o *RoBERTa fine-tuned* e 3.791,50 s para o *LLaMA 3.1 fine-tuned*.

Ainda assim, de modo geral, nenhuma estratégia de ICL supera o desempenho do *fine-tuning* completo, sobretudo quando comparada ao LLM ajustado. Na coleção SST-1, por exemplo, com 60 exemplos, a abordagem *RoBERTa fine-tuned + KNN* atinge 54.9±1.7, configurando empate estatístico com o *RoBERTa fine-tuned* (53.8±1.3). Entretanto, esse resultado é obtido a um custo computacional substancialmente maior (2.061,09 s contra 437,86 s). De forma semelhante, na coleção ACM, com cerca de 30 exemplos, a estratégia *RoBERTa fine-tuned + KNN* (69.0±1.3) atinge desempenho comparável ao *RoBERTa fine-tuned* (70.3±1.4) em termos de efetividade, porém novamente com tempo de execução significativamente superior (5.191,57 s contra 1.043,31 s).

Em síntese, os resultados demonstram que representações vetoriais mais informativas reduzem de forma consistente a lacuna em relação ao *fine-tuning*, aproximando o ICL do desempenho dos modelos totalmente ajustados. Entretanto, observa-se que, nos cenários em que essa aproximação ocorre, o custo computacional pode superar aquele associado ao ajuste e à classificação com modelos previamente treinados, impactando diretamente o *trade-off* entre efetividade e eficiência.

4.4 Escalabilidade e limitações práticas do ICL

As análises anteriores indicam que os ganhos do ICL devem ser interpretados em conjunto com o custo associado à construção e ao processamento dos *prompts*. Embora essa abordagem não envolva treinamento adicional, a inclusão de exemplos rotulados no *prompt* desloca parte do custo para a etapa de inferência. Esse custo decorre, primeiro, da seleção dos exemplos, que pode ser simples, como na escolha aleatória, ou envolver etapas adicionais de geração de representações e recuperação por similaridade. Decorre, também, do tamanho do *prompt*, já que cada exemplo adicional aumenta a quantidade de *tokens* processados pelo LLM em cada classificação. Assim, a escalabilidade do ICL pode ser afetada tanto pela estratégia de seleção adotada quanto pela quantidade de exemplos incluídos no *prompt*, especialmente em cenários com muitos documentos ou textos mais longos.

Outra limitação prática está associada à janela de contexto do modelo e ao consumo de memória durante a inferência. Cada exemplo inserido no *prompt* consome parte do contexto disponível, reduzindo o espaço para o texto a ser classificado e para a resposta do modelo. O caso da coleção ACM, discutido na Seção 4.2, ilustra essa restrição ao mostrar que configurações com muitos exemplos podem tornar-se inviáveis devido a limitações de memória. Dessa forma, a escolha da quantidade de exemplos deve considerar não apenas possíveis ganhos de efetividade, mas também as restrições de contexto, memória e tempo de execução.

5 Conclusão e Trabalhos Futuros

Este estudo investigou o uso de LLMs em CAT, com foco no *in-context learning* como alternativa intermediária entre *zero-shot* e *fine-tuning*. Os resultados confirmam que o *fine-tuning* de SLMs, em especial o RoBERTa, apresenta o melhor equilíbrio global entre efetividade e eficiência [Cunha *et al.*, 2025b], consolidando-se como a opção mais vantajosa sob a perspectiva de *trade-off* custo-benefício. Por outro lado, o ICL com LLMs mostrou-se capaz de reduzir substancialmente a lacuna de desempenho em relação aos modelos totalmente ajustados, especialmente quando associado a estratégias de seleção baseadas em representações vetoriais informativas. Observamos, contudo, que tal aproximação pode implicar aumento relevante no custo computacional, além de depender fortemente das características da coleção, da qualidade das representações adotadas e das limitações impostas pelo tamanho dos *prompts*.

Como direções futuras, destacam-se o desenvolvimento

de métodos de representação mais eficazes e eficientes, a investigação de alternativas ao KNN que promovam maior diversidade na seleção de exemplos e a exploração de estratégias complementares, como a ordenação dos exemplos nos *prompts*, com o objetivo de aprimorar o desempenho sem comprometer a eficiência. Além disso, abordagens híbridas que combinem de maneira mais eficiente os pontos fortes de LLMs e SLMs configuram-se como um caminho promissor para ampliar a efetividade mantendo um equilíbrio adequado entre custo e desempenho.

Declarações complementares

Financiamento

Esta pesquisa foi financiada por: CNPq, Capes, Fapemig, Fapesp, AWS, NVIDIA, CIIA-Saúde e Instituto Nacional de Ciência e Tecnologia em Inteligência Artificial Responsável para Linguística Computacional, Tratamento e Disseminação de Informação (INCT-TILD-IAR; 408490/2024-1).

Contribuições dos autores

G. Prenassi, G. Fonseca, M. A. Gonçalves e L. Rocha contribuíram para a concepção do estudo (*Conceptualization*) e análise formal dos resultados (*Formal analysis*). G. Prenassi foi responsável pelo desenvolvimento dos códigos (*Software*), investigação (*Investigation*) e escrita do rascunho original (*Writing – original draft*). L. Rocha atuou na supervisão da pesquisa (*Supervision*). G. Fonseca e M. A. Gonçalves colaboraram na validação da metodologia (*Validation*). Todos os autores participaram da revisão (*Writing – review & editing*) e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de dados e materiais

Os conjuntos de dados (e/ou softwares) gerados e/ou analisados durante o estudo atual serão feitos mediante solicitação

Outras informações relevantes

Ferramentas de Inteligência Artificial Generativa foram utilizadas como suporte à revisão gramatical do texto. Os autores realizaram uma revisão completa do texto e assumem total responsabilidade pela integridade das informações apresentadas.

Referências

BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., and Reddy, S. (2024). Llm2vec: Large language models are secretly powerful text encoders.

Chandra, M., Ganguly, D., and Ounis, I. (2025). One size doesn't fit all: Predicting the number of examples for in-context learning. In *Advances in Information Retrieval*, pages 67–84, Cham.

Cunha, W., França, C., Fonseca, G., Rocha, L., and Gonçalves, M. A. (2023). An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In *Proceedings ACM SIGIR*, page 665–674. DOI: 10.1145/3539618.3591638.

Cunha, W., Moreo Fernández, A., Esuli, A., Sebastiani, F., Rocha, L., and Gonçalves, M. A. (2025a). A noise-oriented and redundancy-aware instance selection framework. *ACM Trans. Inf. Syst.*, 43(2). DOI: 10.1145/3705000.

Cunha, W., Rocha, L., and Gonçalves, M. A. (2025b). A thorough benchmark of automatic text classification: From traditional approaches to large language models.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Edwards, A. and Camacho-Collados, J. (2024). Language models for text classification: Is in-context learning enough? In *Proceedings of the 2024 LREC-COLING 2024*, pages 10058–10072, Torino, Italia.

Fonseca, G., Cunha, W., Prenassi, G., Gonçalves, M. A., and Rocha, L. C. D. D. (2025). Instance-selection-inspired undersampling strategies for bias reduction in small and large language models for binary text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 9323–9340.

Grattafiori, A. *et al*. (2024). The llama 3 herd of models.

Hochberg, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Kumar, S. and Talukdar, P. (2021). Reordering examples helps during priming-based few-shot learning. In *Findings of ACL-IJCNLP 2021*, pages 4507–4518, Online. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-acl.395.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. (2022). What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114. DOI: 10.18653/v1/2022.deelio-1.10.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Lu, Z., Tian, J., Wei, W., Qu, X., Cheng, Y., Xie, W., and Chen, D. (2024). Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models. In *Findings of ACL 2024*, pages 7841–7864. DOI: 10.18653/v1/2024.findings-acl.467.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317. DOI: 10.1147/rd.14.0309.

OpenAI *et al*. (2024). Gpt-4 technical report.

Prenassi, G. S., Fonseca, G., Reis, D., Cunha, W., Gonçalves, M. A., and Rocha, L. (2025). Um estudo comparativo de estratégias de seleção de exemplos para in-context learning aplicado à classificação automática de texto com grandes modelos de linguagem. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 921–927. SBC.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 EMNLP*.

Xu, C., Xu, Y., Wang, S., Liu, Y., Zhu, C., and McAuley, J. (2024). Small models are valuable plug-ins for large language models. In *Findings of ACL 2024*, pages 283–294. DOI: 10.18653/v1/2024.findings-acl.18.