Um Estudo Comparativo de Estratégias de Seleção de Exemplos para *In-Context Learning* aplicado à Classificação Automática de Texto com Grandes Modelos de Linguagem

Gabriel Prenassi[1], Guilherme Fonseca[2], Davi Reis[1], Washington Cunha[2], Marcos André Gonçalves[2], Leonardo Rocha[1]

[1] Universidade Federal de São João del Rei, Brasil
[2] Universidade Federal de Minas Gerais, Brasil
{prenassigabriel,davireisjeus}@aluno.ufsj.br, lcrocha@ufsj.edu.br
{guilhermefonseca,washingtoncunha,mgoncalv}@dcc.ufmg.br

Resumo. *A Classificação Automática de Texto (CAT) com Grandes Modelos de Linguagem (LLMs) pode ser feita via zero-shot (baixo custo, menor efetividade) ou fine-tuning (alto custo, maior efetividade). Este estudo investiga o in-context learning, abordagem intermediária que insere poucos exemplos no prompt, avaliando estratégias de seleção de exemplos. Comparamos seleção aleatória com métodos baseados em representações vetoriais (TF-IDF, RoBERTa, SBERT e LLM2Vec). Os resultados reforçam que o fine-tuning de Small Language Models (SLMs), como RoBERTa, oferece melhor custo-benefício. Ainda assim, o in-context learning mostra-se promissor, superando o zero-shot em efetividade sem exigir os altos custos do fine-tuning, sobretudo com boas estratégias de seleção.*

Abstract. *Text Classification (TC) with Large Language Models (LLMs) can be performed via zero-shot (low cost, limited effectiveness) or fine-tuning (high cost, high effectiveness). This study explores example selection strategies for the intermediate in-context learning approach, comparing random selection to methods based on vector representations such as TF-IDF, SBERT, and LLM2Vec. Results confirm that fine-tuning Small Language Models (SLMs) offers the best cost-effectiveness trade-off. However, in-context learning emerges as a promising alternative, outperforming zero-shot without the fine-tuning computational cost, particularly when using effective selection strategies.*

1. Introdução

Classificação Automática de Texto (CAT) tem evoluído substancialmente, impulsionada por arquiteturas neurais de Aprendizado Profundo baseadas em *Transformers*, como os modelos de linguagem de primeira geração, tais como BERT e RoBERTa [Devlin et al. 2019, Liu et al. 2019] (aqui denominados *Small Language Models* - SLMs), e, mais recentemente, pelos *Large Language Models* (LLMs), como GPT e Llama [OpenAI et al. 2024, Grattafiori et al. 2024]. Atualmente, os LLMs representam o estado da arte em CAT [Cunha et al. 2025].

LLMs podem ser aplicados em CAT de distintas formas, conforme ilustrado na Figura 1. Na abordagem *zero-shot* (Figura 1a) [Lu et al. 2024, Edwards and Camacho-Collados 2024, Chandra et al. 2025], utiliza-se um LLM pré-treinado via *prompting*, sem adaptação ao domínio, o que é computacionalmente eficiente,

porém de efetividade limitada. Em contraste, o *fine-tuning* (Figura 1c) [Cunha et al. 2025] ajusta o modelo por treinamento supervisionado, elevando a especialização, porém com alto custo computacional. Já o *in-context learning* (Figura 1b) [Lu et al. 2024, Edwards and Camacho-Collados 2024, Xu et al. 2024, Chandra et al. 2025] equilibra custo e efetividade ao enriquecer *prompts* com exemplos, sem ajustar pesos, sendo o foco deste estudo. Especificamente, neste trabalho, investigamos as seguintes perguntas: **(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*?

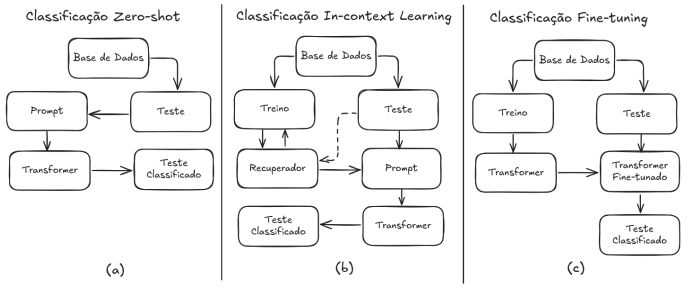

Figura 1. Comparação das abordagens de CAT usando *zero-shot*, *in-context learning* e *fine-tuning*. Em (b), a seta pontilhada indica que, em alguns métodos, são recuperados exemplos semelhantes ao texto a ser classificado.

Para responder às questões, comparamos: (i) *zero-shot*, com Llama 3.1 [Grattafiori et al. 2024]; (ii) *fine-tuning*, com Llama 3.1 (LLM) e RoBERTa [Liu et al. 2019] (SLM), ambos reconhecidos como estado da arte em CAT [Cunha et al. 2023, Fonseca et al. 2024, Cunha et al. 2025]; e (iii) *in-context learning*, explorando diferentes estratégias de seleção de exemplos - desde seleção aleatória até técnicas baseadas em representações vetoriais (TF-IDF [Luhn 1957], RoBERTa, SBERT [Chandra et al. 2025] e LLM2Vec [BehnamGhader et al. 2024]) - e variações na quantidade de exemplos por *prompt*. **Uma ampla análise comparativa de diferentes estratégias de *in-context learning* ainda não foi realizada na literatura, e constitui-se como a principal contribuição deste trabalho.**

Nossos resultados corroboram a literatura [Cunha et al. 2025], indicando que SLMs com *fine-tuning* apresentam a melhor relação entre efetividade e eficiência para CAT (**PP1**). Também demonstramos que o *in-context learning* constitui uma alternativa promissora entre *zero-shot* e *fine-tuning* (**PP2**), com desempenho geralmente melhorado pelo aumento do número de exemplos no *prompt*, embora dependente das características dos dados e de limitações computacionais. Destaca-se que estratégias de seleção baseadas em representações vetoriais proporcionam ganhos relevantes com custo computacional moderado (**PP3**). Ademais, identificamos oportunidades para aprimorar abordagens de *in-context learning* por meio de novos métodos de representação e seleção de exemplos.

2. Trabalhos Relacionados

Na última década, a tarefa de CAT avançou significativamente, impulsionada pelo surgimento de modelos SLMs e LLMs [Cunha et al. 2025]. Nesse cenário, diversos autores vêm propondo estratégias de seleção e uso de exemplos em *prompts*, visando explorar o potencial do *in-context learning* como alternativa intermediária entre *zero-shot* e *fine-tuning*, equilibrando custo computacional e desempenho.

O trabalho de [Liu et al. 2022] propõe a utilização de representações geradas por RoBERTa *fine-tuned* para recuperar exemplos similares ao texto a ser classificado, enriquecendo o *prompt* com contexto relevante. A quantidade de exemplos inseridos varia aleatoriamente conforme o cenário. Em [Lu et al. 2024], os autores introduzem um *framework* em duas etapas: (i) redução do espaço de rótulos por uma técnica de autorredução e (ii) comparações contrastivas par a par entre os rótulos restantes. São utilizados 3 ou 5 exemplos, selecionados aleatoriamente, dependendo do LLM. Já [Edwards and Camacho-Collados 2024] investiga a seleção aleatória de exemplos, inserindo um por classe no *prompt* para comparar *prompting* com *fine-tuning*. Em [Xu et al. 2024], propõe-se a combinação de LLMs com SLMs aplicados ao *fine-tuning* atuando como *plug-ins*, sendo os exemplos escolhidos aleatoriamente, com a quantidade variando por conjunto de dados. Outros dois estudos abordam aspectos distintos. Em [Kumar and Talukdar 2021], investiga-se a influência da ordem dos exemplos no *prompt*, sugerindo que ela impacta a generalização. A seleção dos exemplos é aleatória, e a quantidade não segue critério específico. Já [Chandra et al. 2025] apresenta um método para otimizar a quantidade de exemplos no *prompt*, utilizando um classificador multi-rótulo que mapeia características das instâncias de teste - como *embeddings* e distribuições de rótulos de vizinhos - para prever o número ideal de exemplos, selecionados com base em similaridade via SBERT.

Desses seis trabalhos, apenas [Chandra et al. 2025] foca explicitamente na quantidade de exemplos para o *prompt*. Em quatro estudos, a seleção é aleatória; nos dois restantes, empregam-se modelos SLMs (RoBERTa ou SBERT) para seleção por similaridade. Nosso trabalho se diferencia por investigar simultaneamente dois aspectos comumente tratados separadamente: (i) a variação da quantidade de exemplos no *prompt*; e (ii) diferentes formas de representação vetorial para seleção. Além das abordagens já exploradas na literatura, investigamos representações clássicas como TF-IDF [Luhn 1957] e técnicas mais recentes, como o LLM2Vec [BehnamGhader et al. 2024], que explora a capacidade semântica dos LLMs para gerar *embeddings* informativos.

3. Metodologia Experimental

3.1. Ambiente Experimental

Adotamos o Llama 3.1 como modelo de LLM por ser aberto e apresentar o melhor desempenho em CAT [Cunha et al. 2025]. Nos experimentos de *fine-tuning*, utilizamos a versão Llama-3.1-8B, replicando hiperparâmetros da literatura [Cunha et al. 2025]: taxa de aprendizado inicial de $2e^{-4}$, 4 épocas, *batch size* 4 e *max_len* 256. Para *zero-shot* e *in-context learning*, empregamos Llama-3.1-8B-Instruct, otimizado para *prompting*. Aplicamos quantização em 4 bits em ambos, reduzindo o consumo de memória GPU sem prejuízo relevante [Hu et al. 2021]. Além do Llama, usamos o RoBERTa, referência como SLM em CAT [Cunha et al. 2023, Cunha et al. 2025]. No *fine-tuning* do RoBERTa, seguimos [Cunha et al. 2025] com taxa de aprendizado inicial $5e^{-5}$, máximo de 20 épocas e paciência de 5. Realizamos *grid search* para *max_len* (128, 256) e *batch size* (16, 32). Para *in-context learning*, geramos representações vetoriais com quatro modelos: RoBERTa, RoBERTa *fine-tuned*, SBERT (*all-MiniLM-L6-v2*) [Reimers and Gurevych 2019] e LLM2Vec-Meta-Llama-31-8B-Instruct-mntp [BehnamGhader et al. 2024].

Métrica de Avaliação: Avaliamos a efetividade dos modelos pela Macro-F1, recomendada para cenários com classes desbalanceadas. O custo computacional foi avaliado utili-

zando os tempos de execução (em segundos). Para as estratégias *zero-shot*, considerou-se apenas o tempo de classificação, já que não há treinamento. Nas estratégias de *in-context learning* com representações, incluímos o tempo de geração das representações de treino e teste, além da classificação. A estratégia com representação do RoBERTa *fine-tuned* também inclui o tempo de ajuste dos pesos do modelo. Na abordagem randômica, contabilizamos apenas o tempo de seleção aleatória e de classificação. Já no *fine-tuning* (de LLMs ou SLMs), consideramos o tempo de ajuste dos pesos e de classificação dos exemplos de teste. Os experimentos foram conduzidos com validação cruzada estratificada (*stratified k-fold*), utilizando 10 *folds* em uma instância AWS g5.2xlarge. As diferenças nos resultados foram validadas estatisticamente por meio de um teste-t pareado com nível de confiança de 95%, aplicando-se a correção de Bonferroni [Hochberg 1988].

Bases de Dados: Utilizamos as coleções descritas na Tabela 1. Dimensionalidade refere-se ao tamanho do vocabulário e densidade à média de palavras por documento.

Base de Dados	# Documentos	Dimensionalidade	# Classes	Densidade	Assimetria
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.

3.2. Procedimento Experimental

Avaliamos o desempenho da abordagem *zero-shot* em todos os conjuntos de dados utilizando o *prompt* ilustrado na Figura 2, contendo apenas um exemplo genérico, sem relação com os dados de treinamento, destinado a indicar o formato esperado de resposta (A, B, C etc. são rótulos). Esse exemplo foi mantido fixo para todos os documentos classificados.

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.

Na abordagem *in-context learning*, conduzimos experimentos com inserção progressiva de 10 a 100 exemplos no *prompt*, em intervalos de 10, mantendo a estrutura ilustrada na Figura 2. Na seleção randômica, exemplos foram uniformemente selecionados do conjunto de treinamento. Nas estratégias baseadas em representações vetoriais (TF-IDF ou *embeddings* dos modelos), utilizamos KNN para selecionar os mais similares ao documento alvo, ordenando-os por similaridade decrescente. Cada documento teve um *prompt* específico. Por fim, comparamos os resultados das abordagens *zero-shot* e *in-context learning* ao desempenho obtido por modelos RoBERTa e Llama-3.1-8B após *fine-tuning*.

4. Resultados

Os resultados relacionados às avaliações descritas na seção anterior estão sumarizados nos gráficos da Figura 3. Nos gráficos de (a) a (d), apresentamos os resultados relacionados à efetividade medida por Macro-F1 (eixo y) e nos gráficos de (e) a (h), os resultados

relacionados à eficiência medida pelo tempo (eixo y). Todas as análises são realizadas variando a quantidade de exemplos do treino utilizados nos *prompts* (eixo x), onde $x = 0$ indica resultados relacionados ao *zero-shot*. As linhas azul e laranja representam, respectivamente, os resultados dos modelos Llama-3.1-8B e RoBERTa (SLM), ambos com *fine-tuning* aplicado com todo o conjunto de treinamento e a classificação foi realizada sem o uso de estratégias de *prompting*, o que justifica a ausência de variação da métrica Macro-F1 e Tempo em relação ao número de exemplos utilizados durante a inferência.

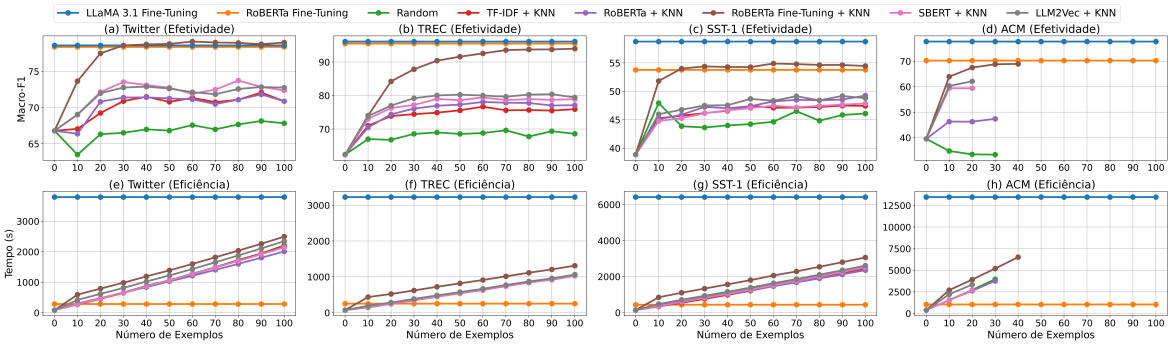

Figura 3. Comparação da Efetividade (Macro-F1) e Eficiência (Tempo) entre abordagens de CAT usando *zero-shot*, *fine-tuning* e variações de *in-context learning*.

PP1 – *Zero-Shot versus Fine-Tuning*: A Figura 3 revela uma diferença marcante na efetividade entre abordagens *zero-shot* e aquelas com *fine-tuning* em LLMs ou SLMs, com estas últimas apresentando desempenho superior. LLMs impõem custo computacional significativamente maior em comparação com abordagens *zero-shot* e SLMs com *fine-tuning*. Os resultados confirmam que SLMs, como RoBERTa, oferecem o melhor equilíbrio entre efetividade e custo computacional [Cunha et al. 2025]. Tal equilíbrio torna os SLMs com *fine-tuning* a escolha mais adequada para tarefas de CAT quando o objetivo é a maximização do balanço (*trade-off*) custo-benefício.

PP2 – *In-Context Learning* entre *Zero-Shot* e *Fine-Tuning*: Para investigar estratégias de *in-context learning*, variamos a quantidade de exemplos nos *prompts*, considerando seis diferentes estratégias, com o objetivo de analisar o impacto dessa variação no desempenho dos modelos em CAT. Os dados apresentados na Figura 3 indicam que não existe um número ótimo universal de exemplos para os *prompts*, variando conforme as características da coleção e a estratégia de seleção adotada. Contudo, observa-se que, em geral, o *in-context learning* aprimora a efetividade da classificação com Llama 3.1 em comparação à abordagem *zero-shot*. Embora o *overhead* associado à seleção de exemplos para *in-context learning* eleve o tempo de processamento em relação ao *zero-shot*, este permanece significativamente inferior ao tempo necessário para o *fine-tuning* dos LLMs, indicando que o *in-context learning* representa uma alternativa promissora entre *zero-shot* e *fine-tuning*. Destaca-se, entretanto, a limitação observada na coleção ACM, onde nenhuma configuração de *in-context learning* com mais de 40 exemplos no *prompt* foi viável devido a restrições de memória GPU, associadas à alta densidade, à elevada dimensionalidade e ao grande número de classes do conjunto. Apesar dos avanços do *in-context learning*, a melhor relação entre efetividade e eficiência continua sendo observada no SLM RoBERTa.

PP3 – Impacto das Representações no Desempenho do *In-Context Learning*: A análise das estratégias de seleção de exemplos destaca sua importância no *in-context learning*, conforme mostrado na Figura 3. Em todas as quatro coleções avaliadas, a seleção randômica teve desempenho consistentemente inferior às abordagens baseadas em representações vetoriais, especialmente com o aumento do número de exemplos nos *prompts*. Na coleção da ACM, onde há maior número de classes, essa diferença foi acentuada, evidenciando que seleções aleatórias tendem a incluir amostras pouco representativas. Isso reforça que *prompts* adaptados a cada documento promovem ganhos de efetividade sem aumentos substanciais de custo computacional. Entre as estratégias vetoriais, o método *RoBERTa Fine-Tuning + KNN* foi o único a se aproximar do desempenho dos modelos RoBERTa e LLaMA 3.1 *fine-tuned*, mesmo com poucos exemplos nos *prompts*, na maioria dos casos. Na coleção Twitter, por exemplo, com apenas 30 exemplos já é possível alcançar a mesma efetividade do LLM com *fine-tuning* a um custo apenas ligeiramente superior ao RoBERTa com *fine-tuning*. Isso sugere que o ajuste da representação ao contexto a ser utilizado é fundamental em abordagens *in-context*.

De modo geral, o método vetorial *RoBERTa Fine-Tuning + KNN* não apresenta ganhos relevantes de efetividade em comparação ao *fine-tuning* isolado e, sobretudo, ao LLM *fine-tuned*. Na coleção SST-1, por exemplo, embora essa abordagem atinja um empate estatístico com o RoBERTa *fine-tuned* utilizando 20 exemplos, isso ocorre com maior custo computacional, e ambas permanecem abaixo da efetividade do LLM. De forma semelhante, na coleção ACM, nenhuma estratégia *in-context* alcança o desempenho do LLM, sendo que a abordagem *RoBERTa Fine-Tuning + KNN*, com cerca de 30 exemplos, consegue igualar o RoBERTa *fine-tuned* em efetividade, mas a um custo substancialmente mais elevado. Esses achados reforçam a necessidade de desenvolver representações vetoriais mais eficazes na recuperação de exemplos relevantes, além de investigar alternativas ao KNN que promovam maior diversidade na seleção dos exemplos nos *prompts*, ampliando o contexto e reduzindo, via *in-context learning*, a lacuna de desempenho frente aos modelos RoBERTa e LLaMA *fine-tuned*.

5. Conclusão e Trabalhos Futuros

Este estudo analisou o uso de LLMs em tarefas de Classificação Automática de Texto (CAT), com ênfase no *in-context learning* como alternativa intermediária entre *zero-shot* e *fine-tuning*. Confirmamos que o *fine-tuning* com SLMs, como RoBERTa, apresenta a melhor relação entre efetividade e eficiência [Cunha et al. 2025]. No entanto, o *in-context learning* com LLMs, aliado a seleções baseadas em representações vetoriais robustas, demonstrou potencial ao reduzir substancialmente a diferença em relação a métodos supervisionados, mesmo com poucos exemplos. Observamos limitações, como sensibilidade aos dados, restrições dos *prompts* e limitações das representações na seleção. Os achados apontam direções para pesquisas futuras, incluindo aprimoramento de técnicas de representação e seleção, bem como abordagens híbridas que combinem de forma eficiente os pontos fortes de LLMs e SLMs.

Agradecimentos

Este trabalho foi apoiado por CNPq, Capes, Fapemig, Fapesp, AWS, NVIDIA, CIIA-Saúde e INCT-TILDIAR (408490/2024-1).

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 of the 46th ACM SIGIR*, SIGIR '23, page 665–674.

Cunha, W., Rocha, L., and Gonçalves, M. A. (2025). 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., Prenassi, G., Cunha, W., Gonçalves, M., and Rocha, L. (2024). Estratégias de undersampling para redução de viés em classificação de texto baseada em transformers. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web*, pages 144–152, Porto Alegre, RS, Brasil. SBC.

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.

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.

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.

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.

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

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.