Explorando o Potencial e a Viabilidade de LLMs Open-Source na Análise de Sentimentos

Breno Braga Neves[1], **Theo Sousa**[1], **Daniel Coutinho**[1], **Alessandro Garcia**[1], **Juliana Alves Pereira**[1]

[1]Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro – RJ – Brazil

Abstract. *Sentiment analysis tools are commonly used in SE to understand developer communication in collaborative environments such as GitHub. As state-of-the-art tools can underperform, newer tools utilizing large language models (LLMs) are being adopted, though they can be computationally expensive. This study evaluates three open-source models: Lllama3, Gemma, and Mistral. Using a GitHub discussion dataset, it explores how these models perform and how prompt engineering influences results. Findings show that these open-source LLMs offer similar performance to state-of-the-art tools, making them viable, cost-effective alternatives. This study also assessed the advantages and limitations of different prompting strategies.*

Resumo. *Ferramentas de análise de sentimentos são amplamente usadas em SE para entender a comunicação de desenvolvedores em ambientes colaborativos, como o GitHub. Como as ferramentas de ponta podem apresentar limitações de desempenho, novos LLMs têm sido adotados, embora sejam computacionalmente caros. Este estudo avalia três modelos open-source: Lllama3, Gemma e Mistral. Utilizando dados de discussões do GitHub, investigamos o desempenho desses modelos e como a engenharia de prompts impacta os resultados. Os resultados indicam que os LLMs open-source oferecem desempenho semelhante às ferramentas de ponta, sendo alternativas viáveis e econômicas. Também analisamos as vantagens e limitações das diferentes estratégias de prompt.*

1. Introdução

No desenvolvimento moderno de software, plataformas colaborativas na nuvem, como GitHub[1] e GitLab[2], são cruciais para permitir desenvolvimento assíncrono e distribuído. Nesse cenário, a eficácia na comunicação torna-se essencial para o sucesso de projetos. Estudos indicam que os sentimentos presentes nessas comunicações podem impactar significativamente o desenvolvimento de software, influenciando desde a introdução de *bugs* até a aceitação ou rejeição de *pull requests* (PRs) [Barbosa et al. 2020, Barbosa et al. 2023, Yu et al. 2015, Tsay et al. 2014].

Portanto, a análise de sentimentos em mensagens trocadas por desenvolvedores pode auxiliar na mitigação de impactos negativos, como desmotivação [Ain et al. 2017]. Uma avaliação recente destacou variações significativas no desempenho de ferramentas de análise de sentimento em mensagens de ES [Coutinho et al. 2024]. Essas variações

[1]https://github.com
[2]https://about.gitlab.com

tornam os LLMs particularmente atraentes, uma vez que, embora ferramentas tradicionais demandem menos recursos computacionais, elas são menos adaptáveis a contextos específicos da ES [Zhang et al. 2023a]. Essa avaliação também indicou um retrabalho significativo para tentar a adaptação dessas ferramentas a diferentes domínios, projetos e/ou comunidades. Em busca de evitar retrabalho, LLMs (*Large Language Models*) se destacam como alternativas [Coutinho et al. 2024], por sua capacidade em entender e incorporar nuances específicas do domínio (e.g. jargões, terminologia técnica) [Brown et al. 2020, Gururangan et al. 2020].

O objetivo, portanto, é avaliar o desempenho das LLMs Llama 3, Gemma, e Mistral em tarefas de análise de sentimentos. Utilizando versões menores desses LLMs, será demonstrado que, quando utilizados em conjunto com técnicas de *prompt engineering*, podem ser tão acessíveis e econômicos quanto ferramentas tradicionais. As principais contribuições desse estudo são: *(i)* Comparação do desempenho de três LLMs open-source (Llama 3, Mistral e Gemma) com cinco ferramentas da literatura na análise de sentimentos, usando um dataset de 1.791 mensagens de 36 projetos [Coutinho et al. 2024]; *(ii)* Análise do impacto de diferentes estratégias de prompt engineering no desempenho dos LLMs open-source e *(iii)* Discussão sobre alternativas viáveis para análise de sentimentos em SE com LLMs menores e open-source.

2. Fundamentação Teórica e Trabalhos Relacionados

2.1. Análise de Sentimentos

A análise de sentimentos estuda a detecção de opiniões, sentimentos e emoções humanas [Hasan et al. 2024]. No contexto da ES, a compreensão dos sentimentos das equipes é essencial, visto que, em trabalhos anteriores, sentimento foi correlacionado com produtividade e qualidade [Graziotin et al. 2014, Graziotin et al. 2015]. Consequentemente, na gestão de projetos de software, a análise de sentimentos pode dar *feedback* valioso ao analisar comunicações textuais, como mensagens de chat e comentários em sistemas de controle de versão. Isso permite a identificação de padrões de sentimentos negativos, possibilitando intervenções proativas [Herrmann and Klünder 2021]. Além disso, em reuniões de projetos, essa análise pode monitorar a dinâmica das interações e detectar comportamentos destrutivos, contribuindo para um ambiente mais colaborativo [Herrmann and Klünder 2021]. Ao manter um ambiente de trabalho positivo, é possível incentivar a inovação e a criatividade nas equipes de desenvolvimento de software. Assim, a análise e manutenção de sentimentos em um ambiente colaborativo não só melhora a comunicação e a colaboração, mas também promove o bem-estar geral das equipes, contribuindo significativamente para o sucesso dos projetos [Ain et al. 2017, Herrmann and Klünder 2021].

2.2. LLMs na Análise de Sentimentos e o Prompt Engineering

A capacidade dos LLMs de capturar nuances sutis e contextos complexos os torna particularmente úteis para a análise de sentimentos em diversos domínios, como avaliações de produtos e interações em redes sociais [Zhan et al. 2024, Niimi 2024, Wei et al. 2022]. [Zhan et al. 2024] demonstrou que técnicas de ajuste fino (*fine-tuning*) podem otimizar os LLMs, alcançando bons resultados em tarefas de análise de sentimentos. Essa capacidade de adaptar os LLMs a tarefas específicas, por meio de técnicas como o *prompt engineering*, melhora ainda mais a precisão e a robustez das análises [Niimi 2024, Xing 2024].

Além disso, a flexibilidade desses modelos em lidar com diferentes contextos e nuances linguísticas permite respostas mais contextualizadas e precisas, o que melhora a interpretação de sentimentos em textos complexos [Zhan et al. 2024, Wei et al. 2022]. Essa flexibilidade é essencial em áreas como marketing, onde a compreensão precisa das emoções e opiniões dos consumidores pode influenciar diretamente as estratégias de negócio [Niimi 2024].

O *prompt engineering*, mencionado anteriormente, refere-se à criação de instruções claras e específicas para orientar os modelos a gerar as respostas desejadas. Prompts bem elaborados ajudam a desambiguar frases e fornecer respostas mais contextualizadas, o que é crucial em discussões de PRs e outras interações em plataformas colaborativas [Coutinho et al. 2024]. Além disso, o *prompt engineering* permite a adaptação dos modelos a diferentes contextos e nuances linguísticas, aumentando a aplicabilidade dos LLMs em variados cenários de análise de sentimentos. Nesse contexto, a inclusão de informações contextuais nos prompts pode melhorar a precisão dos modelos em tarefas que exigem a extração de informações detalhadas e estruturadas [Wei et al. 2022]. Isso é especialmente importante para captar nuances e contextos sutis, essenciais na interpretação correta de sentimentos e intenções expressas, fazendo do *prompt engineering* uma ferramenta valiosa para refinar as respostas dos modelos [Brown et al. 2020, Kaplan et al. 2020, Wei et al. 2022].

3. Metodologia

3.1. Objetivos e Questões de Pesquisa

Este estudo investiga o impacto do *prompt engineering* no desempenho dos LLMs na análise de sentimentos, avaliando-os com e sem essas técnicas, com foco no *F1-Score*. A pesquisa busca identificar as técnicas mais eficazes, demonstrando que LLMs open-source com menor demanda computacional podem ter bom desempenho. Esses LLMs serão comparados com cinco ferramentas tradicionais de análise de sentimentos, para entender suas vantagens e limitações, destacando o potencial dos LLMs open-source. A comparação fornecerá insights sobre as técnicas de prompt engineering mais eficazes, abordando as seguintes questões de pesquisa:

RQ1: Qual é o desempenho dos LLMs open-source na análise de sentimentos utilizando prompts sem técnicas de *prompt engineering?* Essa questão busca, ao estabelecer uma *baseline* inicial sem o uso de técnicas de *prompt engineering*, analisar o comportamento de LLMs open-source em termos de performance para análise de sentimentos.

RQ2: Como diferentes estratégias de *prompt engineering* impactam o desempenho dos LLMs open-source na análise de sentimentos? Nesta questão, foi explorado o impacto de diferentes estratégias de *prompt engineering* no desempenho de LLMs open-source para análise de sentimentos. Isso é feito por meio de uma comparação entre o desempenho desses modelos utilizando o prompt básico (*baseline*) da RQ1 com o desempenho quando aplicadas as técnicas de *prompt engineering*.

RQ3: Como o desempenho dos LLMs open-source se compara com o desempenho dos modelos tradicionais de análise de sentimentos? Essa questão compara o desempenho dos LLMs open-source com as ferramentas tradicionais de análise de sentimentos. O

objetivo é observar e discutir as vantagens e limitações desses LLMs com relação às ferramentas convencionais, identificando cenários onde os LLMs open-source podem oferecer melhorias significativas ou onde as ferramentas ainda são mais eficazes.

3.2. Etapas do Estudo

Passo 1: Obtenção dos Dados. Para este estudo, foi selecionado o *dataset* PRemo, [Coutinho et al. 2024]. Esse *dataset* contém 1.791 mensagens de PRs que tiveram seu sentimento rotulado manualmente por 19 especialistas, incluindo neurocientistas e engenheiros de software. As mensagens utilizadas foram sem nenhum pré-processamento, visto que não foram observadas diferenças nos resultados ao executar esse passo.

Passo 2: Escolha dos LLMs. A seleção das LLMs para as tarefas de análise de sentimento foi baseada em critérios gerais de desempenho, adaptabilidade, e precisão em diferentes cenários. Foi considerado o desempenho dos modelos em tarefas de classificação de sentimento, sua capacidade de ajuste fino com modificações nos prompts, bem como a eficácia ao aplicar técnicas de *prompt engineering*, especialmente em dados limitados ou em idiomas específicos. Além disso, foi levado em conta a capacidade dos modelos de capturar nuances complexas de sentimento e reconhecer emoções em contextos mistos de linguagem.

Com base nesses critérios, foram escolhidos os modelos Llama 3 versão 8b instruct, Gemma 1.1 versão 7b instruct e Mistral 0.2 versão 7b instruct. [Vorakitphan et al. 2024] destacam o bom desempenho do Llama 3 na extração de tríades de sentimentos em cenários de zero e few-shot, enquanto [Jiang et al. 2024] demonstram sua alta eficácia em configurações few-shot com dados limitados no idioma alvo. [Touvron et al. 2023b] confirmam o desempenho elevado do Llama 3 na geração de texto e classificação de sentimento, atribuindo isso à sua capacidade de ajuste fino dos prompts e à necessidade de modificações mínimas. Por outro lado, [Mo et al. 2024] destacam que o Gemma 1.1 7b instruct, quando ajustado, supera modelos como DistilBERT e Llama, alcançando a maior precisão, *recall* e *F1-score*, mostrando-se eficaz na captura de nuances complexas de sentimento. Por fim, [Hou and Lian 2024] mencionam o desempenho competitivo do Mistral 0.2 7b instruct em comparação com modelos como ChatGPT e Llama, enquanto [Siino 2024] evidenciam a eficácia do Mistral 0.2 7b instruct no reconhecimento de emoções em conversas *code-mixed*, alcançando um *F1-score* competitivo com uma estratégia de *few-shot learning*.

Passo 3: Escolha das Técnicas de Prompt Engineering. Esse passo foi iniciado com um prompt básico (*baseline*) como base para comparação. Foram escolhidos prompts em inglês, pois os modelos de linguagem são geralmente mais treinados nesse idioma, proporcionando desempenho superior em relação a outras línguas [Zhang et al. 2023b]. Além disso, a maioria do material de suporte e documentação técnica está em inglês, facilitando o uso e compreensão de *prompt engineering* [Ramesh et al. 2023].

A escolha das técnicas de *prompt engineering* utilizadas neste estudo foi planejada para maximizar o desempenho dos LLMs em tarefas de análise de sentimentos. Visto isso, foram selecionadas diferentes abordagens que se mostraram em trabalhos anteriores capazes de melhorar o raciocínio de LLMs em diferentes tarefas [Wei et al. 2022, Touvron et al. 2023a]. Os prompts analisado incluem ”Prompt One-Shot”, ”Prompt Few-Shot”e ”Prompt Chain of Thought (CoT)”, além dos prompts ”Prompt Simples Zero-

Tabela 1. Prompts utilizados

Nome	Técnica de Prompt Engineering	Prompt dado	Exemplos
Prompt Básico		Classify the sentiment of this GitHub PR message as positive, neutral, or negative.	
Prompt Simples Zero-Shot	O modelo é solicitado a realizar uma tarefa sem exemplos adicionais	Classify the sentiment of this GitHub PR discussion message as positive, neutral, or negative. Use the Shaver emotion model: consider love and joy as positive; anger, sadness, and fear as negative; surprise as context-dependent; and neutral as the absence of any emotion.	
Prompt Complexo Zero-Shot	Fornece uma descrição mais detalhada da tarefa sem exemplos	This message is a part of a discussion from a GitHub PR. Perform sentiment analysis in this message by classifying it as three types of sentiment: positive, neutral, and negative. For this task, utilize the Shaver emotion model, where love and joy (and related emotions) are considered positive, anger, sadness, and fear are considered negative, surprise can be positive or negative depending on the context, and neutral is considered the absense of any emotions.	
Prompt One-Shot	Um exemplo específico é fornecido ao modelo	This message is part of a discussion from a GitHub pull request. Perform sentiment analysis on this message by classifying it as three types of sentiment: positive, neutral, and negative	Message: "This code looks great! Good job." Classification: Positive
Prompt Few-Shot	Apresenta vários exemplos para ajudar o modelo a entender melhor a tarefa	This message is part of a discussion from a GitHub pull request. Perform sentiment analysis on this message by classifying it as three types of sentiment: positive, neutral, and negative	Message: "This code looks great! Good job." Classification: Positive Message: "The code needs some corrections, but it's on the right track." Classification: Neutral Message: "This code is terrible and needs to be redone from scratch." Classification: Negative
Prompt Chain of Thought	Envolve uma série de passos intermediários detalhados para ajudar o modelo a seguir um raciocínio lógico e sistemático	This message is part of a discussion from a GitHub PR. Perform sentiment analysis on this message by classifying it as three types of sentiment: positive, neutral, and negative. Utilize the Shaver emotion model, where love and joy (and related emotions) are considered positive, anger, sadness, and fear are considered negative, surprise can be positive or negative depending on the context, and neutral is considered the absence of any emotions	Message: "The code is functional but can be optimized to improve performance." Reasoning: The message mentions that the code is functional, which is a neutral statement. However, the suggestion for optimization does not express negative emotions, just a constructive suggestion. Classification: Neutral

Shot"e "Prompt Complexo Zero-Shot", que não requerem passos em específico na criação do prompt. A tabela 1 descreve os prompts neste estudo, listando as técnicas de *prompt engineering* utilizadas na criação de cada prompt.

Para a técnica Chain of Thought (CoT), foram utilizados passos intermediários que orientam o modelo a seguir um raciocínio lógico e sistemático. No prompt, o modelo primeiro identifica as partes chave da mensagem, neste caso, *"The code is functional"* como uma afirmação neutra e *"but can be optimized to improve performance"* como uma sugestão construtiva, sem carga emocional negativa. Em seguida, ele relaciona essas partes com as categorias do modelo de emoções de Shaver, concluindo que a mensagem não expressa emoções positivas ou negativas, mas sim uma avaliação objetiva, levando à classificação final como neutro. A partir desse entendimento, espera-se que o modelo replicará essa abordagem em outras análises, mantendo a coerência na classificação dos sentimentos.

Passo 4: Configuração e Utilização dos LLMs. Foi desenvolvido um código em Python utilizando a ferramenta Ollama, que roda localmente os modelos escolhidos no Passo 2. Cada execução da ferramenta analisa as mensagens e as classifica, gerando um arquivo JSON que facilita a análise posterior dos resultados. O *hardware* utilizado para rodar esses modelos inclui um computador com 32GB de RAM, um processador i9 12900k e uma placa de vídeo RTX 3080 com 10GB de VRAM.

Passo 5: Análise dos Dados. A fase final envolve a análise dos dados, onde foi utilizado o *F1-score* para avaliar e comparar o desempenho dos LLMs e dos modelos de análise de sentimentos. O *F1-score* combina a precisão e o *recall*, sendo especialmente útil em

datasets desbalanceados. Foi desenvolvido um código em Python que utiliza pandas e matplotlib para carregar, manipular e limpar os dados em *dataframes*. Em seguida, foram criadas visualizações, como gráficos de barras e linhas, para facilitar a interpretação dos resultados obtidos no Passo 3.

4. Resultados e Discussão

Esta seção apresenta os resultados obtidos a partir da metodologia descrita na seção anterior. A Tabela 2 destaca os principais resultados para cada RQ. Além disso, os prompts estão ordenados em ordem crescente de complexidade.

Tabela 2. Desempenho dos LLMs por prompt

Modelo	Basic	Simples Zero-Shot	Complexo Zero-Shot	One-Shot	Few-Shot	CoT
Gemma	0.53	0.55	0.59	0.56	0.56	**0.59**
Mistral	0.51	0.62	0.61	0.61	**0.62**	0.61
Llama	**0.57**	0.47	0.57	0.48	0.46	0.50

4.1. RQ1: Qual é o desempenho dos LLMs na análise de sentimentos utilizando o prompt sem técnicas de *prompt engineering*?

A Tabela 2 mostra o desempenho dos modelos Llama3, Gemma e Mistral na análise de sentimentos sem uso de técnicas de *prompt engineering*. Observa-se que os modelos não apresentaram variações significativas no desempenho entre si. O desempenho médio apresentado foi de 0.54, onde o Llama3 obteve o melhor desempenho, alcançando um *F1-score* de 0.57, e o desempenho mais baixo foi registrado pelo Mistral, com um *F1-score* de 0.51. Estes resultados estabelecem uma base inicial para comparações futuras na RQ2.

4.2. RQ2: Como diferentes estratégias de *prompt engineering* impactam o desempenho dos LLMs na análise de sentimentos?

A Tabela 2 ilustra a variação de desempenho dos modelos Llama3, Gemma e Mistral com diferentes técnicas de *prompt engineering*. Nota-se que o desempenho do modelo Mistral é o mais consistente, mantendo-se elevado em todas as técnicas de *prompt engineering*, especialmente com o prompt *few-shot*. O modelo Gemma também apresenta desempenho relativamente estável, embora com uma leve tendência de queda em prompts mais complexos. Por outro lado, o modelo Llama3 mostra maior variabilidade e uma queda significativa no desempenho ao usar prompts complexos, como o CoT e o Few Shot.

Esses resultados evidenciam que a eficácia das técnicas de *prompt engineering* varia consideravelmente entre os modelos. A comparação entre os resultados da RQ1 e da RQ2 demonstra que, embora o Llama3 tenha apresentado o melhor desempenho com o prompt básico (RQ1), seu desempenho decai significativamente com prompts mais complexos. Em contraste, Mistral e Gemma mostram melhorias consistentes com o uso de técnicas de *prompt engineering*. Isso sugere que a escolha da técnica de prompt deve ser cuidadosamente ajustada para cada modelo específico, reforçando a importância das estratégias de *prompt engineering* na melhoria da precisão e robustez da análise de sentimentos realizada pelos LLMs.

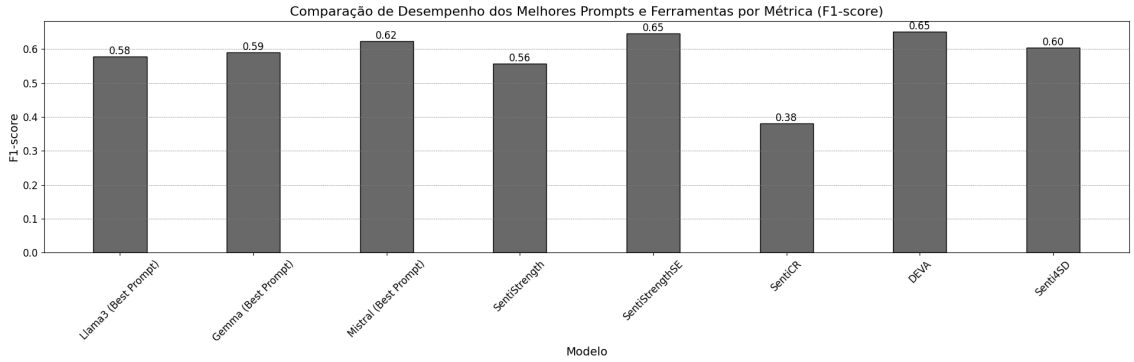

Figura 1. Comparação do desempenho na análise de sentimentos entre LLMs e modelos tradicionais.

4.3. RQ3: Como o desempenho dos LLMs se compara com o desempenho das ferramentas tradicionais de análises de sentimentos?

A Figura 1 compara o desempenho da análise de sentimentos realizada pelos LLMs com as ferramentas tradicionais. Devido ao espaço limitado do artigo, foram apresentados os resultados apenas para a melhor técnica de prompt; o resultado completo pode ser acessado no material complementar [Braga 2024] do artigo. Os resultados mostram que os LLMs, especialmente Mistral e Gemma, com *F1-scores* de 0.624 e 0.590, respectivamente, foram capazes de superar algumas ferramentas de análise de sentimentos.

Notavelmente, o DEVA apresentou o melhor desempenho entre as ferramentas tradicionais, com um *F1-score* de 0.651, destacando-se como a mais precisa. O SentiStrengthSE e o Senti4SD também apresentaram desempenhos relativamente bons, com *F1-scores* de 0.645 e 0.604, respectivamente, posicionando-se próximos aos modelos de LLM. Embora o gráfico na Figura 1 apresente o valor do *F1-score* do SentiStrengthSE como 0.645 (por simplicidade, o valor foi arredondado para duas casas decimais no gráfico). Por outro lado, o modelo SentiCR teve um desempenho significativamente inferior, com um *F1-score* de 0.381.

É importante ressaltar que os valores de *F1-score* ainda são relativamente baixos, indicando que há uma margem considerável para melhorias. Mesmo sem a combinação de técnicas de *prompt engineering*, os resultados foram comparáveis ao desempenho das ferramentas tradicionais de análise de sentimentos. Esses achados indicam que, com o uso adequado de técnicas de *prompt engineering*, LLMs como Mistral e Gemma podem não apenas competir, mas até superar ferramentas tradicionais em algumas situações. No entanto, é necessário continuar aprimorando esses modelos para aumentar sua precisão e confiabilidade, tornando-os soluções mais viáveis para análises de sentimentos em contextos variados e desafiadores. Isso sugere que a exploração de novas técnicas pode levar a um desempenho ainda maior dos LLMs.

4.4. Ameaças de validade

O estudo apresenta algumas ameaças à validade que devem ser consideradas. *(i)* Limitação dos LLMs utilizados, onde modelos com 7 bilhões de parâmetros (Gemma e Mistral) e 8 bilhões de parâmetros (Llama3) foram empregados, influenciados pelo *hardware* disponível; modelos com mais parâmetros poderiam oferecer resultados mais precisos, mas exigem hardware avançado, que não estava disponível. *(ii)* Foco restrito em um

conjunto específico de estratégias de engenharia de prompts, sem explorar outras técnicas ou combinações de estratégias, o que pode ter limitado as descobertas e a identificação de melhores práticas. *(iii)* Viés humano na classificação dos sentimentos, introduzido pela amostragem de dados proveniente do *dataset* PRemo, rotulada manualmente por especialistas, que pode influenciar a percepção do ”certo”e ”errado”nas classificações feitas pelos LLMs.

5. Conclusão

Este estudo investigou o potencial de LLMs na análise de sentimentos em discussões de PRs no GitHub, comparando-os com ferramentas tradicionais de análise de sentimentos. A pesquisa mostrou que os LLMs, especialmente quando combinados com técnicas de *prompt engineering*, podem oferecer desempenho competitivo e, em alguns casos, superior ao das ferramentas.

Os principais achados deste estudo incluem: *(i)* Desempenho variável dos LLMs, com destaque para os modelos Mistral e Gemma, que apresentaram melhorias significativas ao utilizar técnicas de *prompt engineering*, enquanto o Llama3 mostrou desempenho inconsistente, sugerindo que a eficácia das técnicas pode variar entre os LLMs. *(ii)* Comparação com modelos tradicionais, onde os LLMs demonstraram potencial, mas ferramentas como DEVA e SentiStrengthSE ainda superam os LLMs; em particular, o DEVA obteve o melhor desempenho geral, indicando que o uso combinado dessas abordagens pode ser vantajoso. *(iii)* Impacto das técnicas de *prompt engineering*, com a técnica Complex Zero-Shot se destacando como a mais eficaz para melhorar o desempenho dos LLMs, ressaltando a importância de selecionar cuidadosamente as técnicas para maximizar o potencial desses modelos.

Mesmo sem explorar uma ampla gama de técnicas de *prompt engineering*, os resultados foram comparáveis aos das ferramentas tradicionais de análise de sentimentos, com uma diferença máxima de desempenho de apenas 7%. Isso evidencia o potencial promissor dos LLMs, que, com ajustes adequados nas técnicas de *prompt engineering*, podem atingir ou até superar o desempenho das ferramentas tradicionais. Além de serem mais fáceis de configurar, os LLMs oferecem uma alternativa competitiva, sugerindo que a exploração de novas combinações de técnicas pode levar a melhorias significativas. No entanto, desafios persistem para que esses modelos alcancem seu desempenho ideal e sejam amplamente adotados no desenvolvimento de software. Continuar a pesquisa, focando em LLMs mais robustos e no aprimoramento das técnicas de *prompt engineering*, pode resultar em avanços substanciais na análise de sentimentos, beneficiando a comunicação e colaboração em plataformas como o GitHub.

Agradecimentos. Esta pesquisa foi parcialmente apoiada por agências de fomento brasileiras: CAPES (88881.879016/2023-01) e FAPESP (2023/00811-0).

Referências

Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., and Rehman, A. (2017). Sentiment analysis using deep learning techniques: a review. *International Journal of Advanced Computer Science and Applications*, 8(6).

Barbosa, C., Uchôa, A., Coutinho, D., Assunçao, W. K., Oliveira, A., Garcia, A., Fonseca, B., Rabelo, M., Coelho, J. E., Carvalho, E., et al. (2023). Beyond the code: Investiga-

ting the effects of pull request conversations on design decay. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–12. IEEE.

Barbosa, C., Uchôa, A., Coutinho, D., Falcão, F., Brito, H., Amaral, G., Soares, V., Garcia, A., Fonseca, B., Ribeiro, M., et al. (2020). Revealing the social aspects of design decay: A retrospective study of pull requests. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, pages 364–373.

Braga, B. (2024). Complementary material. https://github.com/aisepucrio/llms4s-confmatrixscripts/tree/breno-article. Accessed: setembro/2024.

Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners.

Coutinho, D., Cito, L., Lima, M. V., Arantes, B., Pereira, J. A., Arriel, J., Godinho, J., Martins, V., Libório, P., Leite, L., Garcia, A., Assunção, W. K. G., Steinmacher, I., Baffa, A., and Fonseca, B. (2024). "looks good to me ;-)": Assessing sentiment analysis tools for pull request discussions. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024)*, page 11, Salerno, Italy. ACM.

Graziotin, D., Wang, X., and Abrahamsson, P. (2014). Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ*, 2:e289.

Graziotin, D., Wang, X., and Abrahamsson, P. (2015). How do you feel, developer? an explanatory theory of the impact of affects on programming performance. *PeerJ Computer Science*, 1:e18.

Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Hasan, M. A., Das, S., Anjum, A., Alam, F., Anjum, A., Sarker, A., and Noori, S. R. H. (2024). Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783v2*.

Herrmann, M. and Klünder, J. (2021). From textual to verbal communication: Towards applying sentiment analysis to a software project meeting. In *Leibniz University Hannover*.

Hou, G. and Lian, Q. (2024). Benchmarking of commercial large language models: Chatgpt, mistral, and llama. *Shanghai Quangong AI Lab*. DOI: `https://doi.org/10.21203/rs.3.rs-4376810/v1`.

Jiang, A. Q., Sablayrolles, A., Mensch, A., et al. (2024). The model arena for cross-lingual sentiment analysis: A comparative study in the era of large language models. *arXiv preprint arXiv:2406.19358v1*.

Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models.

Mo, K., Liu, W., Xu, X., Yu, C., Zou, Y., and Xia, F. (2024). Fine-tuning gemma-7b for enhanced sentiment analysis of financial news headlines. *arXiv preprint arXiv:2406.13626*.

Niimi, J. (2024). Dynamic sentiment analysis with local large language models using majority voting: A study on factors affecting restaurant evaluation. *arXiv preprint arXiv:2407.13069*.

Ramesh, K., Sitaram, S., and Choudhury, M. (2023). Fairness in language models beyond english: Gaps and challenges. *arXiv preprint arXiv:2302.12578*.

Siino, M. (2024). Transmistral at semeval-2024 task 10: Using mistral 7b for emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 298–304. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., et al. (2023a). Llama: Open and efficient foundation language models.

Touvron, H., Martin, L., Stone, K., et al. (2023b). Large language models performance comparison of emotion and sentiment classification. *arXiv preprint arXiv:2407.04050v1*.

Tsay, J., Dabbish, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in github. pages 356–366. ACM.

Vorakitphan, V., Basic, M., and Meline, G. L. (2024). Deep content understanding toward entity and aspect target sentiment analysis on foundation models. *Proceedings of the 41st International Conference on Machine Learning*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Xing, F. (2024). Designing heterogeneous llm agents for financial sentiment analysis. *arXiv preprint arXiv:2401.05799*.

Yu, Y., Wang, H., Filkov, V., Devanbu, P., and Vasilescu, B. (2015). Wait for it: Determinants of pull request evaluation latency on github. In *Mining software repositories (MSR), 2015 IEEE/ACM 12th working conference on*, pages 367–371. IEEE.

Zhan, T., Shi, C., Shi, Y., Li, H., and Lin, Y. (2024). Optimization techniques for sentiment analysis based on llm (gpt-3). *arXiv preprint arXiv:2405.09770*.

Zhang, W., Deng, Y., Liu, B., Pan, S. J., and Bing, L. (2023a). Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Zhang, X., Li, S., Hauer, B., Shi, N., and Kondrak, G. (2023b). Don't trust chatgpt when your question is not in english: A study of multilingual abilities and types of llms. *arXiv preprint arXiv:2305.16339*.