

Avaliação de LLMs como Recurso de Apoio ao Aprendizado de Programação Básica

Mika M. dos Santos Júnior¹, Alanna C. da Silva², Diego M. A. Oliveira¹,
Djefferson Maranhão¹, Carlos de Salles Soares Neto¹

¹ Departamento de Informática – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brazil

² Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)
São Luís – MA – Brazil

{kakamarques.0202, alannacostas55, djefferson.maranhao}@gmail.com

{diego.miqueias@discente, carlos.salles@}ufma.br

Abstract. *The growing use of large language models (LLMs) has impacted collaborative platforms such as Stack Overflow, reducing the registration of new questions over the years, which raises doubts about the quality of these responses. Therefore, this study evaluates the semantic quality, through the BERTScore metric, of the capacity of the models GPT-4.1 Mini, Gemini 2.5 Flash, Claude Sonnet 4.6 and DeepSeek-V3 to answer programming questions, using the 90 highest-rated question-answer pairs from the platform as a benchmark. The results show moderate semantic similarity, indicating that these models have potential, but do not yet replace human support in programming learning.*

Resumo. *O crescente uso de grandes modelos de linguagem (LLMs) tem impactado plataformas colaborativas como o Stack Overflow, reduzindo o cadastro de novas perguntas ao longo dos anos, o que gera dúvidas sobre a qualidade dessas respostas. Diante disso, este estudo realiza uma avaliação da qualidade semântica, por meio da métrica BERTScore, da capacidade dos modelos GPT-4.1 Mini, Gemini 2.5 Flash, Claude Sonnet 4.6 e DeepSeek-V3 em responder perguntas de programação, usando os 90 pares de pergunta e resposta mais bem avaliados da plataforma como benchmark. Os resultados mostram similaridade semântica moderada, indicando que esses modelos têm potencial, mas ainda não substituem o suporte humano no aprendizado de programação.*

1. Introdução

O Stack Overflow é uma das plataformas colaborativas de programação mais consolidadas da internet, sendo historicamente uma das principais bases de conhecimento para estudantes e programadores na resolução de dúvidas sobre código.

Com a popularização de grandes modelos de linguagem (LLMs), a plataforma registrou uma queda expressiva no volume de novas perguntas submetidas [del Rio-Chanona et al. 2023], sinalizando uma mudança no comportamento na busca por suporte técnico. Estudantes iniciantes, por exemplo, passaram a recorrer diretamente a ferramentas como ChatGPT e GitHub Copilot para obter explicações de código e resolver

exercícios introdutórios de programação [da Silva Junior et al. 2023]. No contexto educacional, essa mudança de comportamento merece atenção, pois, ao aprender programação, a qualidade da resposta recebida durante o aprendizado é fundamental para consolidar e garantir o entendimento desses estudantes [Denny et al. 2024].

Diante desse cenário, surge uma questão central: as respostas geradas por LLMs são semanticamente equivalentes àsquelas produzidas por especialistas humanos em plataformas colaborativas? Apesar do crescente uso dessas ferramentas como apoio ao aprendizado, a literatura ainda carece de avaliações sistemáticas e quantitativas que comparem a qualidade semântica das respostas automáticas com as produzidas pela comunidade humana, sobretudo no contexto de programação básica em Python.

Nesse sentido, este trabalho avalia a capacidade dos modelos GPT-4.1 Mini, Gemini 2.5 Flash, Claude Sonnet 4.6 e DeepSeek-V3 em responder perguntas de programação em Python, tendo como referência os 90 pares de pergunta-resposta mais bem avaliados do Stack Overflow. A qualidade semântica das respostas geradas é mensurada por meio do BERTScore [Zhang et al. 2019], métrica baseada em *embeddings* contextuais que permite comparações mais robustas do que abordagens tradicionais baseadas em sobreposição de *n*-gramas. Durante a avaliação, identificou-se um viés sistemático relacionado ao tamanho das respostas, para o qual se propõe um controle por quantidade de *tokens* como forma de mitigação. Os resultados obtidos contribuem para orientar estudantes, educadores e pesquisadores quanto ao potencial e às limitações do uso dessas ferramentas como apoio ao aprendizado de programação.

O artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica, abordando os modelos avaliados e as métricas de avaliação semântica utilizadas. A Seção 3 discute os trabalhos relacionados, apresentando estudos com metodologia similar e destacando as diferenças em relação ao presente trabalho. A Seção 4 descreve a metodologia adotada, detalhando a preparação do *Golden Set*, a criação dos *datasets* com as LLMs e o processo de avaliação semântica. A Seção 5 apresenta os resultados obtidos, organizados de acordo com a aplicação ou não do controle de tamanho. A Seção 6 discute e analisa esses resultados de acordo com as questões de pesquisa propostas. Por último, a Seção 7 apresenta as conclusões do estudo e aponta direções para trabalhos futuros.

2. Fundamentação teórica

LLMs são sistemas de processamento de linguagem natural baseados em arquiteturas *transformer*, treinados em volumes massivos de texto com o objetivo de modelar a distribuição probabilística de sequências de palavras [Naveed et al. 2024]. Essa capacidade permite que os modelos realizem tarefas diversas, como responder perguntas, resumir textos e gerar código, sem necessidade de ajuste fino específico para cada domínio.

Os quatro modelos avaliados neste trabalho foram selecionados por serem amplamente adotados pela comunidade, acessíveis por meio de API pública e por representarem diferentes organizações e abordagens de desenvolvimento, permitindo uma comparação diversificada do estado atual dos LLMs de uso geral. O GPT-4.1 Mini [OpenAI 2025], desenvolvido pela OpenAI, é uma versão compacta da família GPT-4.1, equilibrando capacidade de raciocínio e eficiência de inferência. O Gemini 2.5 Flash [Google Cloud 2025], da Google DeepMind, pertence à família Gemini 2.5 e é otimizado para respostas rápidas com janela de contexto estendida. O Claude Sonnet 4.6 [Anthropic 2026], da Anthropic,

integra a família Claude 4 e destaca-se pelo seguimento de instruções e pela qualidade de geração de tarefas de raciocínio. O DeepSeek-V3 [Liu et al. 2024], desenvolvido pela DeepSeek AI, é um modelo de código aberto que alcança desempenho competitivo com modelos proprietários em *benchmarks* de programação e raciocínio lógico.

Para avaliar a qualidade semântica das respostas geradas por esses modelos, este trabalho adota o BERTScore [Zhang et al. 2019] como métrica principal. Métricas tradicionais baseadas em sobreposição de n-gramas, como o BLEU [Papineni et al. 2002] e o METEOR [Banerjee and Lavie 2005], penalizam respostas que expressam o mesmo conteúdo semântico com vocabulário ou estrutura diferentes dos da referência, uma limitação especialmente problemática no domínio de programação, em que existem múltiplas formas igualmente corretas de explicar um conceito. O BERTScore contorna essa restrição ao calcular a similaridade semântica por meio de *embeddings* contextuais gerados por um modelo *encoder* pré-treinado, sendo o *F1-Score* adotado como métrica principal de comparação.

A escolha do *encoder* é determinante para a qualidade do BERTScore. Inicialmente, foi considerado o roBERTa-large [Liu et al. 2019], pré-treinado em 160 GB de texto e reconhecido por sua robustez em tarefas de similaridade semântica em comparação ao BERT tradicional, [Devlin et al. 2019]. No entanto, sua janela de contexto limitada a 512 *tokens* implica o truncamento de respostas mais extensas, comprometendo a representação semântica utilizada pelo BERTScore. Como parte das respostas presentes tanto no *Golden Set* quanto nos *datasets* gerados pelas LLMs ultrapassa esse limite, optou-se pela adoção do ModernBERT-large [Warner et al. 2025] como *encoder*. O modelo suporta janelas de contexto de até 8.192 *tokens*, foi treinado em 2 trilhões de *tokens*, incluindo dados de código-fonte, e é otimizado para GPUs modernas, características que o tornam particularmente adequado para a avaliação de respostas técnicas no domínio de programação.

3. Trabalhos Relacionados

A literatura sobre avaliação de LLMs como ferramentas de apoio ao aprendizado de programação ainda é limitada. Em seus trabalhos, [Denny et al. 2024] discutem os impactos do uso de ferramentas generativas no ensino de computação, argumentando que a qualidade da resposta recebida durante o aprendizado é determinante para a consolidação do conhecimento e que avaliações sistemáticas dessas ferramentas são necessárias para orientar educadores e estudantes. Nessa mesma direção, [da Silva Junior et al. 2023] investigam o comportamento de estudantes que recorrem ao ChatGPT para resolver exercícios introdutórios de programação, identificando tanto benefícios quanto riscos associados à dependência dessas ferramentas sem validação crítica das respostas.

No que diz respeito ao uso do *Stack Overflow* como referência de qualidade, [del Rio-Chanona et al. 2023] documentam a queda no volume de novas perguntas submetidas à plataforma após a popularização dos LLMs, sugerindo uma mudança estrutural na forma como programadores buscam suporte técnico. Esse fenômeno reforça a pertinência de utilizar os pares pergunta-resposta mais bem avaliados da plataforma como *benchmark* de qualidade humana, uma vez que representam o conhecimento coletivo consolidado e validado pela comunidade ao longo dos anos.

Do ponto de vista metodológico, o trabalho mais próximo do presente estudo é o

de [da Cunha et al. 2025], que avalia um sistema *RAG* aplicado a documentos normativos hospitalares em português, utilizando um *Golden Set* de 90 pares pergunta-resposta e o *BERTScore F1* como métrica principal de avaliação da etapa de geração. Os autores concluem que a combinação do modelo de *embedding intfloat/multilingual-e5-small* com o método de reordenação RRF e o gerador Gemini 1.5 Flash oferece o melhor custo-benefício para o domínio avaliado.

No presente estudo, os modelos respondem diretamente às questões sem acesso a contexto externo, diferentemente de [da Cunha et al. 2025], em que as LLMs operam sobre documentos internos de domínio fechado. Além disso, este trabalho emprega o ModernBERT-large como *encoder* do BERTScore em substituição ao roBERTa-large, e identifica um viés sistemático relacionado ao tamanho das respostas de referência, artefato não discutido em trabalhos anteriores que utilizam o BERTScore como métrica de avaliação de LLMs.

4. Metodologia

A metodologia adotada busca avaliar a qualidade semântica das respostas geradas por modelos de linguagem de grande porte (LLMs) frente a respostas humanas de referência na plataforma Stack Overflow, no contexto da Linguagem Python, e pode ser dividida nas seguintes etapas: Preparação do *Golden Set*, Criação dos *datasets* com as LLMs e Avaliação semântica. O fluxograma representando a metodologia aplicada pode ser visualizado na Figura 1.

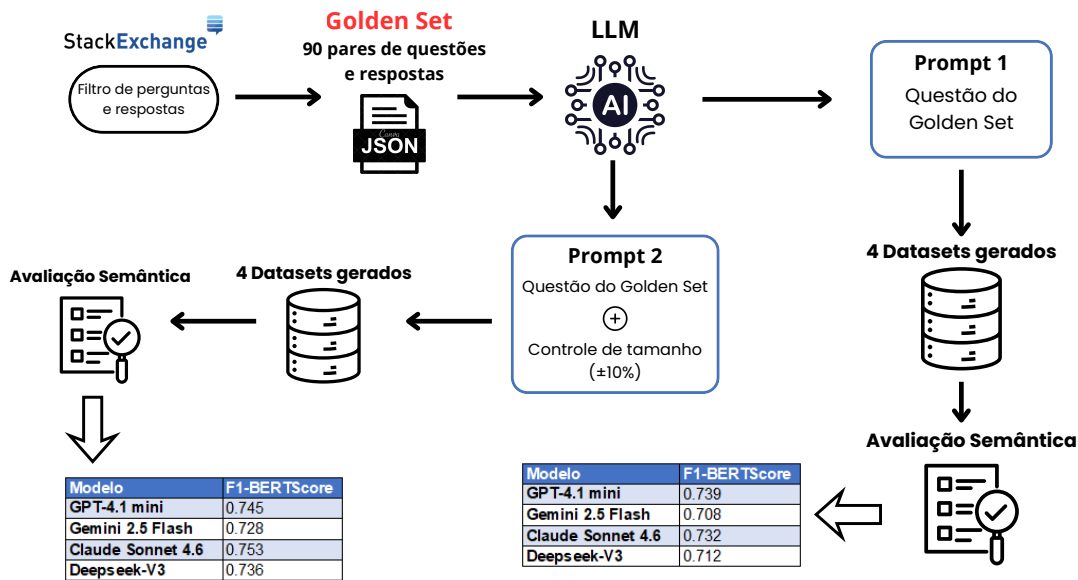


Figura 1. Fluxograma da metodologia

4.1. Preparação do Golden Set

Nesta etapa, utilizou-se a ferramenta Stack Exchange API [Stack Exchange 2026], uma interface pública de acesso programático aos dados públicos da rede de sites Stack Exchange, incluindo o Stack Overflow. O processo de coleta das questões pela API foi

Tabela 1. Exemplos de perguntas e respostas do Golden Set

ID	Questão	Resposta
9	How can I access the index value in a 'for' loop? (Como posso acessar o valor do índice em um loop 'for'?)	Use the built-in function enumerate(): for idx, x in enumerate(xs): print(idx, x) It is non-Pythonic to manually index via for i in range(len(xs)): x = xs[i] or manually manage an additional state variable. Check out PEP 279 for more.
26	How can I delete a file or folder in Python? (Como posso excluir um arquivo ou pasta em Python?)	Use one of these methods: • pathlib.Path.unlink() removes a file or symbolic link. • pathlib.Path.rmdir() removes an empty directory. • shutil.rmtree() deletes a directory and all its contents. On Python 3.3 and below, you can use these methods instead of the pathlib ones: • os.remove() removes a file. • os.unlink() removes a symbolic link. • os.rmdir() removes an empty directory.

automatizado por meio de um *script* desenvolvido em Python, a partir do *endpoint /questions*, utilizando o parâmetro de filtro temático *tagged* para capturar apenas as questões no contexto de programação em Python, além do uso dos parâmetros de ordenação *sort* com o valor “votes” e *order* com o valor “desc”, a fim de ordenar as questões em relação ao *score* líquido em forma decrescente, sem a aplicação de filtros adicionais de qualidade.

No total, foram separados os primeiros 90 pares de pergunta-resposta retornados pela *API* para compor o *Golden Set*, que foi usado de base para todas as etapas de avaliação subsequentes. A amostra de 90 pares foi definida considerando a capacidade de processamento, o custo das APIs das LLMs utilizadas e a necessidade de estabelecer um conjunto representativo e gerenciável para a análise semântica. Cada par foi estruturado em formato JSON, contendo as chaves:

- **ID**: identificador único da questão no *Golden Set*, atribuído conforme a ordem de coleta.
- **Question**: referente a pergunta do par.
- **Answer**: referente a resposta do par.

A Tabela 1 exibe dois exemplos de pares de pergunta-resposta presentes no *Golden Set*. Todo o conjunto está disponível no repositório público¹.

4.2. Criação dos Datasets com as LLMs

Na sequência, as questões do *Golden Set* foram submetidas a quatro LLMs: GPT-4.1 mini [OpenAI 2025], Gemini 2.5 Flash [Google Cloud 2025], Claude Sonnet 4.6 [Anthropic 2026] e Deepseek-V3 [Liu et al. 2024]. Para cada questão, foi enviado um *prompt* composto pelo identificador do par (ID) e pelo enunciado da questão (Question), acompanhado de uma instrução de sistema que define o papel do modelo como especialista em Ciência da Computação e em documentação Python, além de definir regras

¹<https://github.com/Akimkj/SBIE-Artigo>

rígidas de formatação, como proibição de texto fora do JSON e de elementos Markdown, tendo em vista garantir que o modelo retornasse exatamente um JSON válido contendo os campos *id*, *question* e *answer*.

Inicialmente, nenhuma restrição além da formatação Markdown foi imposta às respostas, deixando a cargo de cada modelo determinar o que seria necessário para responder às perguntas fornecidas. Contudo, após a identificação de um viés relacionado ao tamanho das respostas na métrica de avaliação, conforme discutido na Subseção 4.3, foi introduzido um controle de tamanho em *tokens*, tomando como base a resposta de referência do *Golden Set*. Para calcular o tamanho em *tokens* de uma resposta humana, foi implementada uma função de contagem que utiliza o tokenizador nativo de cada modelo: para o GPT-4.1 mini e o Deepseek-V3 foi utilizada a biblioteca *tiktoken* com o tokenizador *cl100k_base*, para o Gemini 2.5 Flash e o Claude Sonnet 4.6 foram utilizados os métodos nativos de contagem de *tokens* das APIs.

Por fim, o valor retornado pelo contador de *tokens* foi usado como parâmetro de tamanho para a resposta do modelo correspondente, com uma tolerância de $\pm 10\%$, de modo que cada LLM gere uma resposta com extensão em *tokens* semelhante à resposta de referência. Durante o processo de geração, foram identificadas respostas com JSON sintaticamente inválido, como texto fora do escopo da estrutura, em especial nas requisições ao Gemini 2.5 Flash. Nesses casos, o *script* foi reexecutado manualmente, reprocessando apenas os IDs cujas respostas não foram salvas corretamente, até que os 90 pares fossem gerados no formato de um JSON válido. Ao final, foram gerados 8 *datasets*, 4 sem controle de tamanho e 4 com controle, cada um contendo 90 pares de pergunta-resposta com a mesma estrutura abordada na Subseção 4.1.

4.3. Avaliação semântica

Para avaliar a qualidade das respostas geradas pelas LLMs, foi adotado o BERTScore [Zhang et al. 2019] como métrica principal. O BERTScore compara a similaridade semântica entre uma sentença candidata e uma de referência por meio de *embeddings* contextuais, que geram diferentes representações vetoriais para a mesma palavra dependendo do contexto, permitindo capturar o significado semântico de forma robusta, em comparação a modelos de avaliação baseada na sobreposição de n-gramas como o BLEU [Papineni et al. 2002] e METEOR [Banerjee and Lavie 2005].

A escolha do modelo de *encoder* é uma etapa crucial para o cálculo do BERTScore. Inicialmente, o roBERTa-large [Liu et al. 2019] foi escolhido por causa da sua eficiência em diversidade lexical e flexibilidade em variações linguísticas, além de ser um modelo pré-treinado em 160GB de texto, tornando-o mais robusto que o modelo BERT [Devlin et al. 2019] tradicional na tarefa de similaridade semântica. No entanto, o roBERTa-large possui uma janela de contexto limitada a 512 *tokens*, o que implica no truncamento de respostas mais extensas e, conseqüentemente, compromete a representação semântica utilizada pelo BERTScore. Como parte das respostas presentes tanto no *Golden Set* quanto nos *datasets* ultrapassam esse limite, optou-se pela troca do encoder pelo *ModernBERT-large* [Warner et al. 2025], que possui uma janela de contexto de até 8192 *tokens*. A escolha do ModernBERT-large também se deve por ser um modelo otimizado para GPUs modernas e pelo seu treinamento em 2 trilhões de *tokens*, incluindo dados de código, o que o torna apropriado para contextos de programação.

Durante a aplicação do BERTScore, foi identificado um viés sistemático nas pontuações associadas às questões com respostas de referência mais curtas. Para investigar esse comportamento, os itens do *Golden Set* foram reordenados de forma crescente em relação ao comprimento em *tokens* das respostas, e o BERTScore foi recalculado e visualizado em forma de um *heatmap*. O gráfico revelou uma concentração de pontuações F1 na extremidade esquerda, correspondente às questões de respostas mais curtas. Para reduzir esse efeito, foi adicionado o controle de tamanho descrito na Subseção 4.2. Os resultados desse procedimento são apresentados na Seção 5.

Para resumir os resultados, foram calculadas a média, mediana e desvio padrão das três componentes do BERTScore (*precision*, *recall* e F1), para cada um dos *data-sets* gerados, sendo o *F1-Score* adotado como métrica principal de comparação entre os modelos, por representar a média harmônica entre o *precision* e o *recall*. Por fim, com o objetivo de verificar se as diferenças entre os modelos são estatisticamente significativas, foi aplicado o teste de Wilcoxon pareado com correção de Holm para múltiplas comparações. Além disso, foi calculada a matriz de correlação entre as pontuações F1 dos modelos, para investigar se os padrões de variação entre as questões são similares.

5. Resultados

Esta seção apresenta os resultados obtidos após a aplicação do BERTScore sobre os conjuntos gerados pelas LLMs, organizados de acordo com o controle de tamanho abordado na Subseção 4.2.

5.1. Resultados sem Controle de Tamanho

A Figura 2 apresenta o *heatmap* comparativo dos *scores* F1 obtidos pelos quatro modelos avaliados para cada uma das 90 questões do *Golden Set*, sem a aplicação do controle de tamanho. Cada linha representa uma LLM e cada coluna representa uma questão, sendo a cor proporcional ao score obtido, variando do vermelho, passando pelo amarelo e indo até o verde.

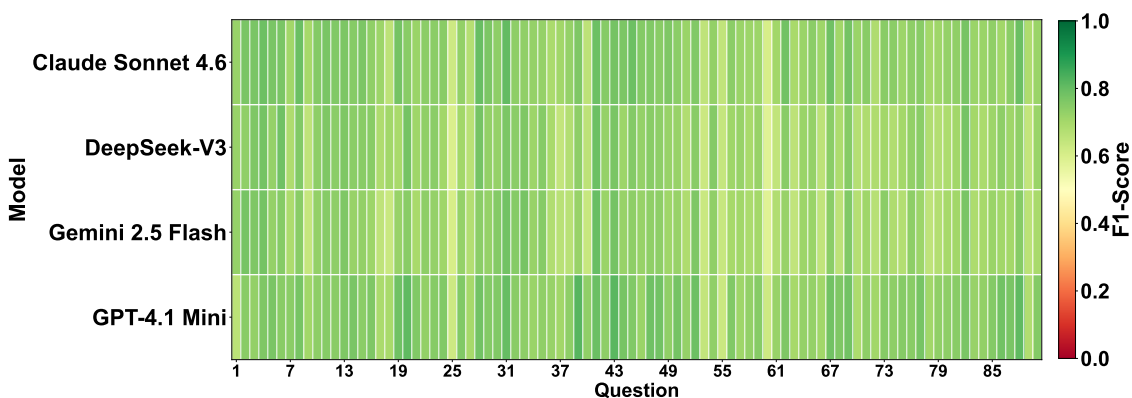


Figura 2. Distribuição do BERTScore F1 por Modelo e Questão, Sem Controle de Tamanho

No processo de inspeção do viés sistemático de tamanho do BERTScore abordado na Subseção 4.3, foi gerada a Figura 3, outro *heatmap* comparativo dos *scores* F1 das respostas geradas pelas LLMs sem controle de tamanho, mas ordenado de forma crescente em relação ao tamanho das respostas.

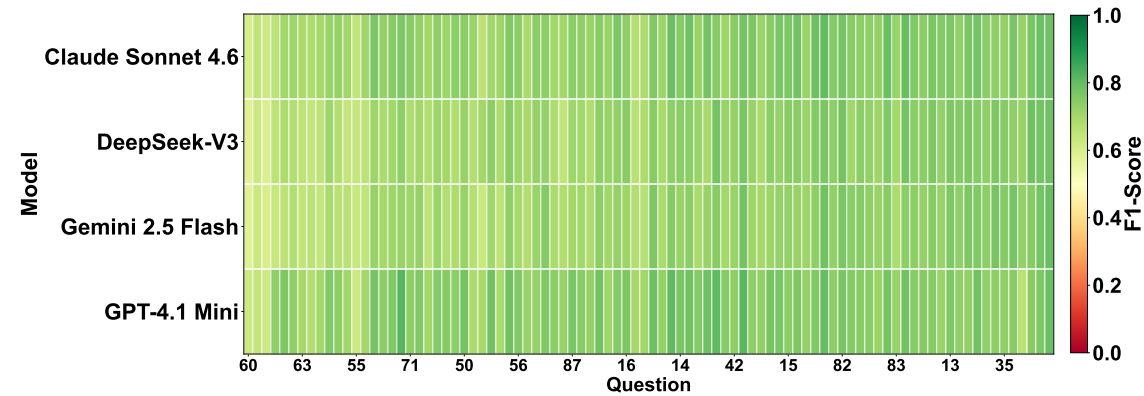


Figura 3. Distribuição do BERTScore F1 por Modelo: Questões Ordenadas por Comprimento da Resposta de Referência (Sem Controle de Tamanho)

5.2. Resultados com Controle de Tamanho

A Figura 4 exibe o heatmap comparativo dos *scores* F1 ordenados de forma crescente em relação ao comprimento das respostas de referência correspondentes, mas desta vez com o controle de tamanho de $\pm 10\%$ em tokens.

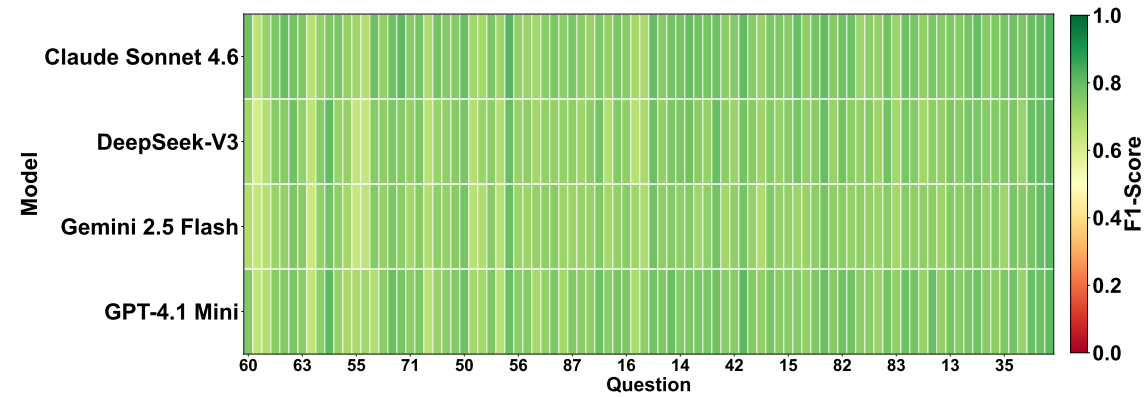


Figura 4. Distribuição do BERTScore F1 por Modelo: Questões Ordenadas por Comprimento da Resposta de Referência (Com Controle de Tamanho)

5.3. Comparativo dos Resultados com e sem Controle de Tamanho

A Tabela 2 apresenta as estatísticas descritivas do BERTScore para os quatro modelos avaliados, considerando os cenários com e sem controle de tamanho das respostas. No grupo sem controle de tamanho, o OpenAI GPT-4.1 Mini obteve o maior F1 médio, com valor de 0,7396, seguido pelo Claude Sonnet 4.6, com 0,7322. Em seguida aparecem o DeepSeek V3, com 0,7130, e o Gemini 2.5 Flash, com 0,7088.

A Tabela 3 mostra a variação do BERTScore F1 após a aplicação do controle de tamanho, calculada como a diferença entre o F1 no cenário controlado e o F1 no cenário sem controle. Os resultados indicam que todos os modelos apresentaram ganho médio positivo de F1 com o controle de tamanho.

O maior ganho médio foi observado no DeepSeek V3, com aumento de 0,0236 no F1, seguido pelo Claude Sonnet 4.6, com 0,0214, e pelo Gemini 2.5 Flash, com 0,0196.

Tabela 2. Métricas dos BERTScore com e sem controle de tamanho

Modelo	Sem controle de tamanho					Com controle de tamanho				
	F1			Precision	Recall	F1			Precision	Recall
	Média	Mediana	DP	Média	Média	Média	Mediana	DP	Média	Média
OpenAI GPT-4.1 Mini	0.7396	0.7438	0.0388	0.7329	0.7497	0.7459	0.7487	0.0359	0.7490	0.7433
Gemini 2.5 Flash	0.7088	0.7169	0.0422	0.6616	0.7666	0.7284	0.7322	0.0344	0.7163	0.7415
Claude Sonnet 4.6	0.7322	0.7378	0.0382	0.7022	0.7682	0.7536	0.7605	0.0345	0.7509	0.7570
DeepSeek V3	0.7130	0.7196	0.0413	0.6678	0.7688	0.7365	0.7383	0.0383	0.7408	0.7329

Tabela 3. Variação do F1-BERTScore

Modelo	$\Delta F1$			Núm. Melhorias (<i>n</i>)	Proporção (<i>n</i> / <i>N</i>)
	Média	Mediana	DP		
DeepSeek V3	0.0236	0.0163	0.0323	71	78,88%
Claude Sonnet 4.6	0.0214	0.0147	0.0327	69	76,66%
Gemini 2.5 Flash	0.0196	0.0150	0.0297	65	72,22%
OpenAI GPT-4.1 Mini	0.0063	0.0052	0.0283	56	62,22%

O OpenAI GPT-4.1 Mini apresentou o menor ganho médio, de 0,0063. Além disso, o número de questões com melhoria também reforça esse padrão. O DeepSeek apresentou melhoria em 71 das 90 questões, correspondendo a 78,88% do total. O Claude melhorou em 69 questões, o Gemini em 65 e o OpenAI em 56.

A Tabela 4 apresenta a matriz de correlação dos valores de BERTScore F1 entre os modelos, separadamente para os cenários com e sem controle de tamanho. No cenário sem controle de tamanho, a maior correlação foi observada entre DeepSeek V3 e Gemini 2.5 Flash, com coeficiente de 0,9083. Também foram observadas correlações elevadas entre Claude e Gemini, com 0,8692, e entre Claude e DeepSeek, com 0,8272. O OpenAI apresentou correlações menores com os demais modelos, especialmente com Gemini, com 0,7266, sugerindo um padrão de variação distinto nesse aspecto.

Tabela 4. Matriz de correlação do BERTScore F1

Modelo	Sem controle de tamanho				Com controle de tamanho			
	Claude	DeepSeek	Gemini	OpenAI	Claude	DeepSeek	Gemini	OpenAI
Claude Sonnet 4.6	1.0000	0.8272	0.8692	0.7914	1.0000	0.8339	0.7999	0.7987
DeepSeek V3	0.8272	1.0000	0.9083	0.7442	0.8339	1.0000	0.8769	0.8350
Gemini 2.5 Flash	0.8692	0.9083	1.0000	0.7266	0.7999	0.8769	1.0000	0.7952
OpenAI GPT-4.1 Mini	0.7914	0.7442	0.7266	1.0000	0.7987	0.8350	0.7952	1.0000

A Tabela 5 apresenta as comparações pareadas entre os modelos por meio do teste de Wilcoxon, com correção de Holm para múltiplas comparações. No grupo sem controle de tamanho, o OpenAI GPT-4.1 Mini foi favorecido nas comparações contra Claude, DeepSeek e Gemini. Em relação ao Claude, a diferença média foi de 0,0074 em favor do OpenAI, com 56 vitórias contra 34 e p-valor corrigido de 0,0202. A vantagem do OpenAI foi ainda mais expressiva em relação ao DeepSeek e ao Gemini, com diferenças médias de 0,0266 e 0,0308, respectivamente, ambas com p-valor corrigido inferior a 0,001.

No grupo com controle de tamanho, o padrão se altera. O Claude Sonnet 4.6

passa a ser o modelo favorecido nas comparações contra DeepSeek, Gemini e OpenAI. A diferença média em relação ao OpenAI foi de 0,0077, com 58 vitórias contra 32 e p-valor corrigido de 0,0008. Em relação ao DeepSeek e ao Gemini, as diferenças foram maiores, de 0,0171 e 0,0252, respectivamente, ambas estatisticamente significativas.

Tabela 5. Comparações pareadas por Wilcoxon entre modelos, com correção de Holm.

Grupo	Comparação	Δ F1 médio	Vitórias	p_{Holm}	Favorecido
Sem controle	Claude vs DeepSeek	0.0192	72–18	< 0.001	Claude
Sem controle	Claude vs Gemini	0.0234	78–12	< 0.001	Claude
Sem controle	Claude vs OpenAI	-0.0074	34–56	0.0202	OpenAI
Sem controle	DeepSeek vs Gemini	0.0042	47–43	0.0318	DeepSeek
Sem controle	DeepSeek vs OpenAI	-0.0266	17–73	< 0.001	OpenAI
Sem controle	Gemini vs OpenAI	-0.0308	15–75	< 0.001	OpenAI
Com controle	Claude vs DeepSeek	0.0171	72–17	< 0.001	Claude
Com controle	Claude vs Gemini	0.0252	85–5	< 0.001	Claude
Com controle	Claude vs OpenAI	0.0077	58–32	0.0008	Claude
Com controle	DeepSeek vs Gemini	0.0081	61–29	0.0002	DeepSeek
Com controle	DeepSeek vs OpenAI	-0.0094	31–58	0.0002	OpenAI
Com controle	Gemini vs OpenAI	-0.0175	18–72	< 0.001	OpenAI

Observação: Δ F1 médio representa a diferença média entre o primeiro e o segundo modelo da comparação. As vitórias indicam o número de questões em que cada modelo obteve maior BERTScore F1. Os p-valores foram ajustados pelo método de Holm.

6. Discussão e Análise dos resultados

Os resultados da métrica BERTScore dos quatro modelos submetidos revelam um panorama entre os dois contextos experimentais. No cenário sem controle, a média de pontuações F1 varia na faixa de 0,7088 a 0,7396, enquanto a introdução do controle de tamanho elevou as pontuações F1 para a faixa de 0,7284 a 0,7536. Outro ponto a observar é o baixo desvio padrão, entre 0,034 e 0,042, indicando que as LLMs mantêm uma forte consistência na geração das respostas. Entretanto, apesar dessa consistência, a magnitude das médias sugere que a semântica das respostas humanas usadas como *benchmark* foram capturados pelos modelos de modo parcial, com divergências ainda notáveis em relação às referências. Esse resultado indica que, embora os modelos apresentem estabilidade na escrita, eles ainda enfrentam limitações para reproduzir integralmente a qualidade semântica e o refinamento técnico das respostas humanas de referência.

Observa-se que, no cenário sem controle, os modelos Claude, DeepSeek e Gemini apresentaram valores médios de *recall* superiores aos de *precision*. Esse resultado sugere que, sem limitação de tamanho, esses modelos tendem a recuperar maior quantidade de conteúdo semanticamente relacionado às respostas de referência, mas com menor precisão média. O OpenAI, por sua vez, apresentou maior equilíbrio entre *precision* e *recall*, o que contribuiu para seu maior F1 médio nesse grupo.

Com a introdução do controle de tamanho, a ordenação dos modelos se altera e os valores de F1 aumentam em todos os casos, confirmando empiricamente o viés sistemático evidenciado na Figura 3 e identificado na Subseção 4.3. O impacto foi mais expressivo nos modelos com maior desequilíbrio entre *precision* e *recall*, como Claude,

DeepSeek e Gemini, enquanto o OpenAI mostrou-se menos sensível à restrição de tamanho, conforme detalhado na Tabela 3.

Nos dois cenários experimentais, o Gemini 2.5 Flash obteve a menor média. Uma hipótese para esse comportamento se deve ao fato de que a API do Gemini retornou, na maioria das requisições, um JSON inválido, além de ser o único modelo que não seguiu a restrição de não usar elementos Markdown em suas respostas, sugerindo que o modelo apresentou maior dificuldade em aderir às regras descritas na instrução de sistema, ao contrário dos demais modelos, que aderiram às restrições definidas.

A matriz de correlação apresentada na Tabela 4 revela que as correlações foram positivas e elevadas em ambos os grupos, indicando que os modelos tendem a apresentar padrões semelhantes de variação entre as questões. Isto é, questões que resultam em maior F1 para um modelo também tendem a produzir maior F1 para os demais. Esse padrão sugere que certas questões são intrinsecamente mais desafiadoras para todos os modelos, independentemente da arquitetura utilizada.

Considerando que as 90 questões foram avaliadas por todos os modelos, o teste de Wilcoxon pareado com correção de Holm, cujos resultados estão apresentados na Tabela 5, é adequado para comparar o desempenho de forma controlada, questão a questão. Os resultados confirmam estatisticamente as diferenças observadas entre os modelos. No grupo sem controle, a OpenAI foi a mais favorecida em todas as comparações com os demais modelos, com p-valores corrigidos inferiores a 0,001. Já no grupo com controle, o Claude passa a ser o modelo mais favorecido em todas as comparações, incluindo em relação ao modelo da OpenAI, com p-valor corrigido de 0,0008, reforçando que a inversão na ordenação dos modelos não é resultante de variação aleatória.

Quanto à questão central do estudo, que investiga se as LLMs são capazes de gerar respostas semanticamente equivalentes às produzidas por humanos e avaliadas como as melhores no contexto da programação, os resultados não permitem confirmar essa equivalência. Embora as pontuações de similaridade indiquem que os modelos testados apresentam potencial como ferramentas de apoio à aprendizagem de programação básica, as respostas geradas não alcançaram, em geral, o mesmo grau de proximidade semântica das respostas mais bem avaliadas no Stack Overflow.

Esse resultado, entretanto, deve ser interpretado considerando as limitações do referencial e da métrica adotados. As respostas do Stack Overflow foram utilizadas como referência por terem sido validadas pela comunidade, mas não representam necessariamente o melhor padrão pedagógico para o ensino de programação básica. Além disso, o BERTScore mede a similaridade semântica entre textos, sem avaliar sua correção técnica. Assim, uma resposta incorreta, mas próxima da referência, pode receber uma pontuação elevada, enquanto uma resposta correta, porém formulada de maneira distinta, pode obter uma pontuação menor. Os resultados, portanto, indicam proximidade semântica, e não correção técnica ou qualidade pedagógica.

7. Conclusão

Este trabalho avaliou a qualidade semântica de respostas geradas por LLMs para questões de programação básica em Python, usando BERTScore em comparação com respostas humanas do Stack Overflow. O GPT-4.1 Mini obteve o melhor desempenho no cenário

sem controle de tamanho; após a normalização por *tokens*, o Claude Sonnet 4.6 alcançou os maiores valores de F1. Em ambos os casos, os resultados indicam similaridades semânticas moderadas, evidenciando que as LLMs ainda não são capazes de oferecer qualidade semântica equivalente a respostas humanas disponíveis em plataformas colaborativas. Ainda assim, os modelos demonstram potencial como ferramentas de apoio ao aprendizado de programação, mas não substituem plenamente o conhecimento coletivo da comunidade.

Esses resultados levantam uma questão relevante: se as LLMs não apresentam uma qualidade semântica de excelência em relação às respostas humanas do Stack Overflow, por que a plataforma registra queda no volume de novas perguntas? Uma hipótese plausível é que a migração dos usuários seja motivada não pela qualidade das respostas, mas pela praticidade da interação, as LLMs oferecem respostas imediatas e contextualizadas, reduzindo o esforço de busca. Esse comportamento reforça a importância de avaliações como a conduzida neste estudo para orientar estudantes e educadores sobre as limitações dessas ferramentas.

Como contribuição metodológica adicional, o estudo identificou e mitigou empiricamente um viés sistemático do BERTScore em relação ao comprimento das respostas: a normalização por *tokens* elevou as pontuações médias e reduziu o desvio padrão, indicando que diferenças de extensão distorcem a avaliação semântica. Esse achado é relevante para estudos que usam a métrica para comparar modelos generativos com referências humanas.

Como trabalhos futuros, pretende-se investigar se a integração de sistemas de Geração Aumentada por Recuperação (RAG) ao processo de geração das LLMs pode aumentar a similaridade semântica e a qualidade das respostas, especialmente ao fornecer informações relevantes sobre o domínio da programação. Pretende-se também ampliar o escopo da avaliação por meio de um conjunto mais abrangente de questões, da inclusão de outras linguagens de programação e da adoção de métricas complementares ao BERTScore. Por fim, pretende-se realizar avaliações com especialistas e estudantes, a fim de examinar aspectos não avaliados neste estudo, como a correção técnica, a clareza das explicações e a qualidade pedagógica das respostas.

Disponibilidade de Artefatos

Os *datasets* gerados neste estudo, incluindo o *Golden Set* e as respostas produzidas pelos modelos avaliados, estão disponíveis em: <https://github.com/Akimkj/SBIE-Artigo>.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Brasil – Código de Financiamento 001. Também conta com financiamento no âmbito do projeto “Wanda: Card Games para Ensino de Lógica de Programação”, processo 88887.180797/2025-00 contemplado pelo Edital nº 3/2025 CAPES “InovaEducação”

Referências

Anthropic (2026). Claude sonnet 4.6. <https://www.anthropic.com/news/claude-sonnet-4-6>. Acesso em: 15 mar. 2026.

- Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- da Cunha, M. V., Silveira, M. R., Santana, B. S., Freitas, L. A., and Corrêa, U. B. (2025). Optimizing and evaluating a retrieval-augmented generation system for normative document retrieval in hospital settings. In *Brazilian Symposium on Multimedia and the Web (WebMedia)*, pages 385–393. SBC.
- da Silva Junior, S. M., de Freitas, R. A. B., de Moraes, M. A. C., and Costa, D. L. V. (2023). Chatgpt no auxílio da aprendizagem de programação: Um estudo de caso. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1375–1384. SBC.
- del Rio-Chanona, M., Laurentsyeve, N., and Wachs, J. (2023). Are large language models a threat to digital public goods? evidence from activity on stack overflow. *arXiv preprint arXiv:2307.07367*.
- Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., and Sarsa, S. (2024). Computing education in the era of generative ai. *Communications of the ACM*, 67(2):56–67.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Google Cloud (2025). Gemini 2.5 flash. <https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/2-5-flash>. Acessado em: 18 dez. 2025.
- Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. (2024). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- 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. *arXiv preprint arXiv:1907.11692*.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- OpenAI (2025). Gpt-4.1 mini. <https://developers.openai.com/api/docs/models/gpt-4.1-mini>. Acessado em: 27 mar. 2026.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Stack Exchange (2026). Stack Exchange API documentation. <https://api.stackexchange.com/docs>. Acessado em: 23 dez. 2025.
- Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., et al. (2025). Smarter, better, faster, longer:

A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.