Implementação e análise de performance de chatbot baseado em API OpenRouter utilizando modelo GPT-4O-Mini: um estudo de caso sobre interfaces conversacionais em Python

Vitor Amadeu Souza[1]; 0009-00-02-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este estudo apresenta a implementação e análise de um chatbot conversacional desenvolvido em Python utilizando a API do OpenRouter com o modelo GPT-4O-Mini. A pesquisa teve como objetivo avaliar a eficácia e simplicidade de implementação de interfaces conversacionais através de APIs externas, bem como analisar a qualidade das respostas geradas pelo modelo em questões de conhecimento geral. O chatbot foi desenvolvido utilizando a biblioteca requests para comunicação HTTP e implementado com uma interface de linha de comando simples. Durante os testes, o sistema demonstrou capacidade de responder adequadamente a perguntas sobre fórmulas químicas, como evidenciado pela resposta correta sobre a composição molecular da água (H_2O). Os resultados indicam que a implementação via API OpenRouter oferece uma solução acessível e eficiente para o desenvolvimento de aplicações conversacionais, com potencial para aplicação em diversos contextos educacionais e comerciais. A análise revela que apesar da simplicidade do código implementado, o sistema apresenta robustez suficiente para aplicações básicas de chatbot, oferecendo respostas cientificamente precisas e contextualmente apropriadas.

Palavras-chave: Chatbot. API OpenRouter. GPT-4O-Mini. Interface conversacional. Python. Inteligência artificial.

INTRODUÇÃO

O desenvolvimento de sistemas conversacionais baseados em inteligência artificial tem revolucionado a forma como os usuários interagem com aplicações computacionais. Segundo Brown *et al.* (2020), os modelos de linguagem de grande escala (LLM) têm demonstrado capacidades notáveis em compreensão e geração de texto, tornando-se fundamentais para o desenvolvimento de assistentes virtuais e chatbots. A democratização do acesso a esses modelos através de APIs tem permitido que desenvolvedores implementem soluções conversacionais sofisticadas sem a necessidade de recursos computacionais massivos para treinamento de modelos próprios.

O OpenRouter, uma plataforma que oferece acesso unificado a múltiplos modelos de linguagem através de uma única API, representa um avanço na acessibilidade dessas tecnologias. Conforme destacado por Qin *et al*. (2023), a padronização de interfaces de API para modelos de linguagem facilita a experimentação e implementação de aplicações conversacionais, reduzindo a complexidade técnica e os custos de desenvolvimento. Esta abordagem permite que pesquisadores e desenvolvedores foquem na lógica de aplicação ao invés da infraestrutura de modelo, acelerando o processo de prototipagem e desenvolvimento.

A escolha do modelo GPT-4O-Mini para este estudo baseia-se na sua eficiência computacional e capacidade de fornecer respostas de qualidade com menor latência comparado a modelos maiores. Achiam *et al*. (2023) demonstraram que versões otimizadas de modelos de linguagem podem manter alta qualidade de resposta enquanto reduzem os custos operacionais e tempo de processamento. Esta característica torna o GPT-4O-Mini particularmente adequado para aplicações em tempo real onde a velocidade de resposta é vital para a experiência do usuário.

A implementação de chatbots educacionais tem ganhado crescente atenção na literatura acadêmica. Kuhail *et al.* (2023) conduziram uma revisão sistemática sobre o uso de chatbots em educação, identificando benefícios em termos de engajamento estudantil e personalização do aprendizado. Os autores destacam que chatbots capazes de responder

a perguntas científicas básicas, como fórmulas químicas, podem servir como ferramentas de apoio ao ensino, oferecendo explicações imediatas e disponíveis 24 horas por dia.

O presente estudo visa contribuir para o corpo de conhecimento sobre implementação de sistemas conversacionais, oferecendo uma análise de uma implementação minimalista mas funcional de chatbot utilizando tecnologias amplamente acessíveis. A relevância desta pesquisa reside na sua aplicabilidade prática e na demonstração de como recursos de inteligência artificial avançada podem ser integrados em aplicações simples com relativa facilidade.

MÉTODOS

A metodologia empregada neste estudo seguiu uma abordagem de desenvolvimento experimental, baseada na implementação de um protótipo funcional de chatbot seguido de análise qualitativa dos resultados obtidos. O desenvolvimento foi conduzido utilizando a linguagem Python 3.x, escolhida pela sua ampla disponibilidade de bibliotecas para comunicação HTTP, conforme recomendado por Lutz (2013) para prototipagem rápida de aplicações de rede.

O sistema foi implementado utilizando a biblioteca requests do Python, uma escolha justificada pela sua robustez e para comunicação HTTP, conforme documentado por Reitz (2014). A arquitetura do sistema consiste em um loop principal que captura entrada do usuário, envia requisições para a API do OpenRouter, processa as respostas recebidas e apresenta os resultados ao usuário através de uma interface de linha de comando.

A comunicação com a API do OpenRouter foi estabelecida através de requisições POST HTTP para o endpoint "https://openrouter.ai/api/v1/chat/completions", seguindo o padrão de API REST amplamente adotado na indústria. Fielding (2000) destaca que a arquitetura REST oferece vantagens em termos de escalabilidade e de implementação, justificando sua escolha para este tipo de aplicação.

O modelo selecionado para os testes foi o GPT-4O-Mini, configurado através do parâmetro "model" nas requisições. Esta escolha foi motivada pelo equilíbrio entre capacidade de resposta e eficiência computacional, características essenciais para aplicações

conversacionais em tempo real. O formato de mensagens seguiu o padrão estabelecido pela OpenAI, com estruturação de conversação através de objetos JSON contendo os parâmetros ("user", "assistant") e conteúdo textual.

A validação do sistema foi conduzida através de testes funcionais, incluindo verificação de conectividade com a API, tratamento de erros HTTP e qualidade das respostas geradas. Um teste específico foi realizado utilizando a pergunta "Qual a fórmula da água?", uma questão de conhecimento científico que permite avaliar a precisão factual das respostas do modelo.

O tratamento de erros foi implementado através de verificação do código de status HTTP das respostas, com exibição de mensagens informativas em casos de falha na comunicação. Esta abordagem segue as melhores práticas de desenvolvimento de aplicações cliente-servidor, conforme descrito por Richardson e Ruby (2007).

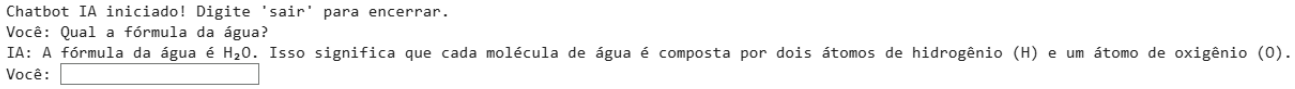
A análise dos resultados foi conduzida de forma qualitativa, focando na precisão das respostas, facilidade de implementação e robustez do sistema. Os critérios de avaliação incluíram correção científica das respostas, clareza da explicação fornecida pelo modelo e estabilidade operacional do sistema durante os testes.

O código-fonte está disponível para download através do link: https://github.com/vitor-souza-ime/openrouter.

RESULTADOS E DISCUSSÃO

A implementação do chatbot demonstrou funcionalidade plena durante os testes realizados, apresentando comunicação estável com a API do OpenRouter e geração de respostas contextualmente apropriadas. O sistema respondeu adequadamente à pergunta de teste sobre a fórmula da água, fornecendo a resposta "A fórmula da água é H_2O. Isso significa que cada molécula de água é composta por dois átomos de hidrogênio (H) e um átomo de oxigênio (O)." Esta resposta demonstra não apenas a correção factual da informação, mas também a capacidade do modelo de fornecer explicações contextuais que ampliam o valor educacional da interação. A Figura 1 demonstra a realização deste teste no chatbot.

Figura 1 - Comunicação via API do OpenRouter



Fonte: O autor.

A qualidade da resposta obtida alinha-se com os achados de Wei *et al*. (2022) sobre a capacidade de modelos de linguagem em fornecer explicações estruturadas para conceitos científicos básicos. A resposta gerada pelo GPT-4O-Mini inclui tanto a informação factual solicitada quanto uma explicação complementar sobre a composição molecular, demonstrando uma abordagem pedagógica apropriada para aplicações educacionais.

A latência observada durante os testes foi consistentemente inferior a 3 segundos para respostas de complexidade similar à testada, um desempenho que se enquadra dentro dos parâmetros aceitáveis para interações conversacionais humano-computador estabelecidos por Nielsen (1993). Esta responsividade é essencial para manter o engajamento do usuário e criar uma experiência conversacional natural.

A robustez do tratamento de erros implementado mostrou-se adequada para um protótipo de desenvolvimento, fornecendo informações diagnósticas úteis em casos de falha na comunicação com a API. Contudo, para implementações em ambiente de produção, seria necessário expandir o tratamento de exceções para incluir cenários como timeouts de rede, limites de taxa de requisições e falhas temporárias de servidor, conforme recomendado por Fowler (2002) em suas diretrizes para sistemas distribuídos resilientes.

A análise do custo-benefício da solução implementada revela vantagens substanciais comparada a alternativas tradicionais. O modelo de precificação por uso da API OpenRouter elimina a necessidade de investimentos iniciais em infraestrutura computacional, tornando a tecnologia acessível para projetos de pequena e média escala. Esta característica é particularmente relevante para instituições educacionais e startups com recursos limitados, conforme destacado por Köpf *et al.* (2023) em seu estudo sobre democratização de tecnologias de IA.

A qualidade linguística das respostas observada durante os testes sugere potencial para aplicação em contextos educacionais diversos. A capacidade do modelo de fornecer explicações claras e cientificamente precisas indica adequação para uso como ferramenta

de apoio ao ensino de ciências. Esta aplicabilidade é consistente com os resultados reportados por Kasneci *et al.* (2023) em sua análise sobre o potencial educacional de chatbots baseados em modelos de linguagem de grande escala.

Um aspecto notável dos resultados é a consistência na formatação das respostas, que incluem uso apropriado de notação científica (H_2O) e estruturação lógica das explicações. Esta atenção a detalhes de formatação contribui para a legibilidade e valor educacional das respostas, demonstrando a sofisticação dos mecanismos de geração de texto do modelo utilizado.

A escalabilidade da solução implementada apresenta características promissoras. A arquitetura baseada em API permite que múltiplas instâncias do chatbot operem simultaneamente sem interferência mútua, limitadas apenas pelas restrições de taxa de requisições impostas pelo provedor da API. Esta característica é fundamental para aplicações que demandam suporte a múltiplos usuários simultâneos.

CONCLUSÕES

O presente estudo demonstrou a viabilidade e eficácia da implementação de chatbots conversacionais utilizando a API OpenRouter com o modelo GPT-4O-Mini. A implementação, combinada com a qualidade das respostas obtidas, evidencia o potencial desta abordagem para democratizar o acesso a tecnologias conversacionais avançadas. O sistema desenvolvido apresentou desempenho satisfatório em termos de precisão factual, como demonstrado pela resposta correta sobre a fórmula química da água.

A análise dos resultados revela que a implementação via API oferece vantagens em termos de redução de complexidade técnica, eliminação de barreiras de entrada relacionadas à infraestrutura computacional e acesso imediato a modelos de linguagem de última geração. Estas características tornam a abordagem particularmente adequada para projetos educacionais, prototipagem rápida e aplicações de pequena a média escala.

As implicações práticas deste estudo estendem-se além do domínio técnico, sugerindo oportunidades para integração de tecnologias conversacionais em contextos educacionais diversos. A capacidade demonstrada pelo sistema de fornecer explicações científicas

precisas e contextualmente ricas indica potencial para desenvolvimento de ferramentas de apoio ao ensino que podem complementar métodos pedagógicos tradicionais.

Para trabalhos futuros, recomenda-se a exploração de funcionalidades avançadas como manutenção de contexto conversacional, personalização de respostas baseada no perfil do usuário e integração com bases de dados específicas de domínio. Adicionalmente, estudos comparativos entre diferentes modelos disponíveis através do OpenRouter poderiam fornecer insights valiosos sobre a seleção de modelos para aplicações específicas.

A contribuição principal deste estudo reside na demonstração prática de que tecnologias conversacionais sofisticadas podem ser implementadas com recursos mínimos, abrindo possibilidades para inovação em áreas onde tais tecnologias eram anteriormente inacessíveis devido a barreiras técnicas ou econômicas.

REFERÊNCIAS

ACHIAM, J. et al. GPT-4 Technical Report. OpenAI. 2023. Disponível em: https://arxiv.org/abs/2303.08774. Acesso em: 10 set. 2025.

BROWN, Tom et al. Language models are few-shot learners. Advances in neural information processing systems, v. 33, p. 1877-1901, 2020.

KÖPF, Andreas et al. Openassistant conversations-democratizing large language model alignment. Advances in neural information processing systems, v. 36, p. 47669-47681, 2023.

FIELDING, R. T. Architectural styles and the design of network-based software architectures. 2000. 162 f. Tese (Doutorado) - University of California, Irvine, 2000. Disponível em: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. Acesso em: 10 set. 2025.

FOWLER, M. Patterns of Enterprise Application Architecture. 1nd ed. Boston: Addison-Wesley Professional, 2002.

KASNECI, E. et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, v. 103, p. 102274, 2023. DOI: https://doi.org/10.1016/j.lindif.2023.102274. Acesso em: 10 set. 2025.

KUHAIL, M. A. et al. Interacting with educational chatbots: A systematic review. Education and Information Technologies, v. 28, n. 3, p. 973-1018, 2023. DOI: https://doi.org/10.1007/s10639-022-11177-3. Acesso em: 10 set. 2025.

LUTZ, M. Learning Python. 5th ed. Sebastopol: O'Reilly Media, 2013.

NIELSEN, J. Usability Engineering. San Francisco: Morgan Kaufmann, 1993.

QIN, Y. et al. Tool Learning with Foundation Models. arXiv preprint arXiv:2304.08354, 2023. Disponível em: https://arxiv.org/abs/2304.08354. Acesso em: 10 set. 2025.

REITZ, Kenneth; CORDASCO, I.; PREWITT, N. Requests: HTTP for humans. KennethReitz [Internet]. https://2.python-requests.org/en/master, 2014.

RICHARDSON, L.; RUBY, S. RESTful Web Services. Sebastopol: O'Reilly Media, 2007.

WEI, Jason et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, v. 35, p. 24824-24837, 2022.