

OBI-UAN: Um Agente para Auxiliar nos Estudos da Olimpíada Brasileira de Informática

Zairo Bastos, Raylander Marques, Gabriel Rudan, Marlon Duarte, Wellington Franco¹

¹Universidade Federal do Ceará (UFC), Fortaleza – CE – Brasil

{zairo.vianahd, raylandermarques, gabrielrudan, marlongduarte}@alu.ufc.br
wellington@crateus.ufc.br

Abstract. We present OBI-UAN, an educational tool based on Large Language Models (LLMs). Using a question bank from previous editions of the Brazilian Computing Olympiad (OBI), the tool organizes questions by topic and difficulty level, following OBI's own guidelines and leveraging Prompt Engineering techniques. Additionally, OBI-UAN allows users to submit their solutions. By relying on a database of correct answers, the system uses semantic similarity retrieval techniques to estimate how closely the user's response matches known solutions. With the support of Retrieval-Augmented Generation (RAG) techniques, it then provides personalized hints to guide the improvement of the user's answer. The system's architecture includes efficient data structuring and indexing to enable fast queries and contextualized feedback.

Resumo. Apresentamos a OBI-UAN^{1 2}, uma ferramenta educacional baseada em modelos largos de linguagem (LLMs). Utilizando um banco de questões das edições anteriores da OBI, a ferramenta organiza-as por tema e nível de dificuldade, de acordo com as próprias diretrizes da OBI, com o uso de técnicas de Engenharia de Prompt. Além disso, a OBI-UAN permite que os usuários submetam suas soluções. Por meio de uma base de respostas corretas, utiliza técnicas de recuperação por similaridade semântica para estimar o quanto a resposta do usuário se aproxima de soluções já conhecidas, e, com apoio de técnicas de RAG, fornece dicas personalizadas para guiar o aprimoramento da resposta. A arquitetura do sistema contempla a estruturação e indexação eficiente dos dados para permitir consultas rápidas e feedback contextualizado.

1. Introdução

A Olimpíada Brasileira de Informática (OBI) é uma competição nacional que desempenha um papel significativo na formação de estudantes da área de Tecnologia da Informação (T.I.), ao incentivá-los a desenvolver habilidades fundamentais em matemática e teoria da computação por meio da resolução de problemas [da Silva Pereira et al. 2019]. No entanto, apesar de sua relevância, ainda são limitadas as ferramentas de apoio aos estudantes de T.I., o que dificulta o estudo autônomo, especialmente quando esse estudo se baseia nas questões da própria OBI [de Lima and de Menezes 2024].

¹Repositório: <https://github.com/zairobastos/obiuan>

²Vídeo de demonstração: <https://youtu.be/YCTlikL30p8>

A organização da OBI disponibiliza uma aplicação gratuita com recursos úteis, como o calendário de provas, ementas, materiais de estudo, gabaritos, sugestões de compiladores e, principalmente, um banco de questões anteriores com a possibilidade de avaliação quanto à assertividade da resolução proposta pelo estudante. Atualmente, as questões são organizadas apenas por ano e modalidade, sem qualquer classificação por tema, assunto ou nível de dificuldade. Essa limitação compromete a criação de planos de estudo mais personalizados e eficientes [Teixeira and Alliprandini 2013]. Além disso, embora a plataforma informe se uma resposta está certa ou errada, ela não fornece explicações ou orientações que ajudem o aluno a compreender os erros cometidos e aprimorar suas soluções.

Para superar essas limitações, propomos a OBI-UAN, uma aplicação web baseada em modelos largos de linguagem (LLMs) e técnicas de recuperação de informações. Uma ferramenta que provê a organização das questões por tema e nível de dificuldade, a geração de *feedbacks* automáticos baseados em similaridade semântica.

Este artigo está organizado em três seções adicionais. A Seção 2 apresenta os trabalhos relacionados, a Seção 3 descreve os detalhes do desenvolvimento da OBI-UAN, e a Seção 4 reúne as considerações finais deste trabalho.

2. Trabalhos Relacionados

Diversas plataformas acadêmicas estão disponíveis para estudo de programação, como *LeetCode*³, *HackerRank*⁴, *Exercism*⁵, *CodeSignal*⁶ e *Codility*⁷. Essas ferramentas oferecem um amplo banco de questões, geralmente categorizadas em níveis, além de possibilitarem que os usuários submetam suas soluções e recebam avaliações automáticas. Apesar disso, todas essas plataformas são majoritariamente voltadas ao público internacional e utilizam o idioma inglês. Outro ponto é que nenhuma dessas ferramentas é especializada no contexto da OBI, uma competição nacional que possui características próprias, tanto no estilo das questões quanto nas competências cobradas. A seguir, discutimos trabalhos recentes que investigam o uso de LLMs na resolução e classificação de problemas de programação e como essas abordagens fundamentam parte do desenvolvimento da OBI-UAN.

O trabalho de [MELO et al. 2023] avaliou o desempenho dos modelos de linguagem ChatGPT-3.5 e Bard na resolução de 83 problemas de programação de diferentes níveis de complexidade, obtidos de plataformas online como o *LeetCode*. Os resultados revelaram que o ChatGPT-3.5 solucionou 20 questões e o Bard 18 das 83 submetidas, com taxas de vereditos “*Accepted*” de 24,1% e 14,9%, respectivamente, indicando um baixo percentual de acerto geral. O ChatGPT-3.5 obteve sucesso principalmente em questões de nível mais fácil, enquanto o Bard apresentou uma alta porcentagem de “*Compilation Error*”, o que não era esperado para um LLM. Conclui-se que os LLMs ainda têm limitações em raciocínio lógico complexo, como reforça o trabalho de [Wang et al. 2024].

Já em [SOUZA 2024], o autor avaliou a eficácia do Chat GPT 3.5 e do Mistral Le

³<https://leetcode.com/>

⁴<https://www.hackerrank.com/>

⁵<https://exercism.org/>

⁶<https://codesignal.com/>

⁷<https://www.codility.com/>

Chat na resolução de 100 questões de lógica da OBI, modalidade iniciação. As questões, provenientes das edições de 2022 e 2023, foram submetidas em texto, com uma única tentativa para cada LLM, evitando questões com imagens. Os resultados mostraram que o Chat GPT 3.5 acertou 23% das questões, enquanto o Mistral Le Chat obteve um desempenho superior, com 36% de acertos. Ambos os modelos demonstraram dificuldade em problemas lógicos, com o Chat GPT 3.5 apresentando má interpretação de questões com a expressão “exceto” e ambos retornando múltiplas alternativas onde apenas uma era correta. A pesquisa conclui que ambos os LLMs são insuficientes para resolver problemas de lógica da OBI, falhando em compreender a complexidade de contextos e abstrações cruciais para a resolução, o que reforça o uso de outras técnicas de aprimoramento dos modelos para isso, como evidenciado por [Yoshida et al. 2024].

No estudo de [Silva et al. 2024], as autoras realizaram um mapeamento sistemático sobre o uso de IA Generativa no ensino de programação, analisando 51 artigos publicados entre 2021 e 2024. Os trabalhos foram classificados em cinco categorias, com destaque para a avaliação de LLMs como o ChatGPT, presente em 49% dos estudos. Os resultados apontam que essas ferramentas têm bom desempenho em tarefas introdutórias, mas enfrentam limitações em problemas mais complexos. Também mostraram potencial para gerar exercícios e fornecer *feedback* personalizado, contribuindo para a motivação dos alunos.

O principal diferencial da OBI-UAN é justamente esse foco: uma plataforma construída inteiramente em língua portuguesa, baseada nas questões da OBI, e pensada para apoiar estudantes brasileiros interessados em melhorar suas habilidades em lógica, programação e resolução de problemas no contexto dessa competição. Além disso, a OBI-UAN inova ao combinar LLMs com técnicas de RAG para oferecer não só a categorização inteligente das questões por tema e dificuldade, mas também a geração de instruções para a solução da questão e *feedbacks* baseados na comparação semântica entre as respostas do usuário e uma base de soluções corretas.

3. OBI-UAN

A ferramenta OBI-UAN foi desenvolvida com o objetivo de auxiliar estudantes da área de Tecnologia da Informação na preparação para a OBI. A aplicação funciona como um guia de estudos, organizando as questões da competição conforme os níveis de dificuldade definidos pela própria comissão organizadora. A seguir, apresentamos a arquitetura da ferramenta, detalhando sua concepção e funcionamento.

3.1. Arquitetura da OBI-UAN

A arquitetura da OBI-UAN é composta por quatro agentes principais, cada um responsável por uma etapa do processo de classificação e rotulação das questões da OBI:

1. **Agente de Rotulação de Assuntos:** Responsável por rotular o tema central de cada questão, classificando-as em: Fundamentos de Programação, Fundamentos de Matemática e Algoritmos e Estruturas de Dados;
2. **Agente de Rotulação Temática:** Após a rotulação inicial, um segundo agente analisa a questão e define, com base na ementa da competição, a qual grupo de subassunto e o tema a que ela pertence. Esse agente também determina o nível da questão segundo a divisão oficial da OBI: Júnior, Nível 1, Nível 2 ou Sênior;

3. **Agente de Nível de Dificuldade:** Responsável por classificar o grau de dificuldade da questão em uma das três categorias: fácil, médio ou difícil;
4. **Agente de Feedback:** Por fim, outro agente é responsável por, considerando o algoritmo que responde a questão, criar um passo-a-passo de como aquele aluno pode aprimorar sua solução para alcançar o objetivo.

Todos os agentes utilizam o mesmo modelo de linguagem base, o GPT-4o, sendo diferenciados apenas pelos seus respectivos *prompts*, elaborados com técnicas de Engenharia de *Prompt*⁸. Essa abordagem permite que cada agente desempenhe funções especializadas, sem a necessidade de múltiplos modelos distintos. Categorizar e classificar as questões permite que os estudantes organizem seus estudos de maneira mais eficiente, focando em questões compatíveis com seu nível atual de conhecimento e priorizando os temas mais recorrentes nas edições anteriores da OBI.

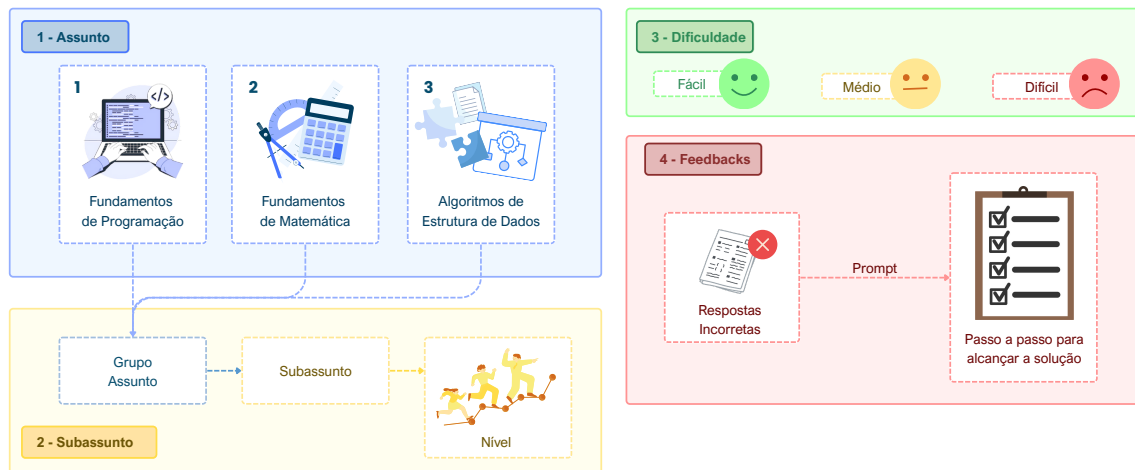


Figura 1. Agentes atuantes da ferramenta.

Para o desenvolvimento da ferramenta OBI-UAN, foram empregadas tecnologias que viabilizam tanto a análise automatizada das questões quanto a interação eficiente com o usuário final. A biblioteca *pandas*⁹ foi empregada para a manipulação, organização e estruturação dos dados brutos das questões da OBI. A classificação automatizada das questões em diferentes categorias foi realizada com o auxílio de modelos de linguagem natural. Utilizou-se especificamente o modelo *gpt-4o-mini*¹⁰. Esse modelo foi responsável por executar as tarefas de rotulação de assuntos, identificação do grupo de subassunto e subassunto, determinação do nível da OBI e atribuição do grau de dificuldade.

Para a interface gráfica, utilizou-se o framework *Streamlit*¹¹, voltado a aplicações web interativas com foco em ciência de dados. Para a comparação entre a resposta do estudante e a resposta correta, foi utilizada a técnica de *embeddings*. Nesta etapa, adotou-se o modelo *Qwen/Qwen3-Embedding-0.6B*¹², disponível na biblioteca *Transfor-*

⁸Os *prompts* utilizados pelos agentes estão disponíveis em: <https://github.com/zairobastos/obiuan/blob/main/prompts.py>

⁹<https://pandas.pydata.org/>

¹⁰<https://platform.openai.com/docs/models/gpt-4o-mini>

¹¹<https://streamlit.io/>

¹²<https://huggingface.co/Qwen>

mers¹³. O modelo gera vetores densos para representar o conteúdo, possibilitando comparar códigos, inclusive de linguagens diferentes, com base na similaridade vetorial.

3.2. Funcionamento do OBI-UAN

O funcionamento da OBI-UAN foi projetado para ser simples, direto e acessível a estudantes em processo de aprendizagem. A Figura 2 ilustra o fluxo básico de acesso às questões, dividido em dois passos principais.

No **Passo 1**, o usuário interage com a *Tela Inicial* da aplicação, que apresenta uma breve descrição sobre a OBI e sobre a própria ferramenta. Nesta tela, o estudante encontra, à esquerda, um painel de navegação com uma barra lateral que organiza as questões em caixas de seleção, categorizadas por temas e organizadas também por nível de dificuldade. Essa organização visa auxiliar o aluno na construção de um plano de estudo direcionado de acordo com seus interesses ou necessidades.

Ao selecionar uma das questões disponíveis, o sistema direciona o usuário para o **Passo 2**, correspondente à *Tela da Questão*. Nessa etapa, o estudante pode ler o enunciado completo do problema, escrever sua solução no campo de resposta fornecido e, em seguida, submeter essa resposta por meio do botão de envio.

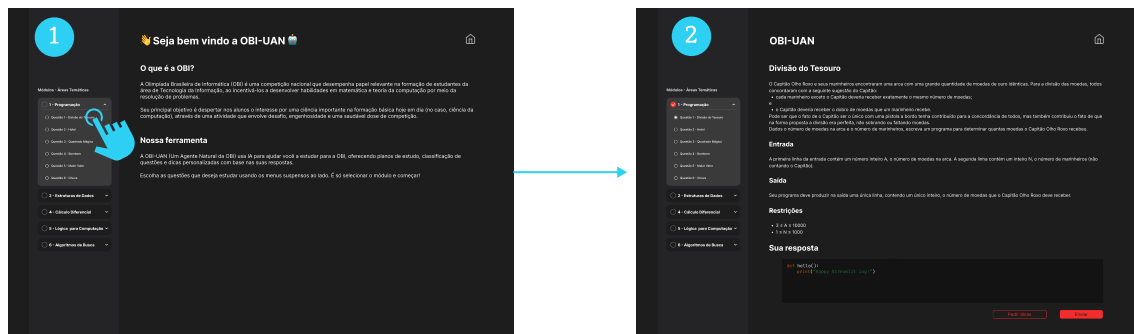


Figura 2. Fluxo de uso da OBI-UAN: Tela Inicial e Tela da Questão

Após o envio, a aplicação processa a resposta, realizando uma análise semântica para verificar sua proximidade em relação às soluções corretas presentes na base, redirecionando para a Tela de Resultado (**Passo 3**). Além disso, o estudante também pode recorrer a um *feedback* textual que orienta-o sobre possíveis ajustes ou melhorias, acessando a Tela de *Feedback* (**Passo 4**).

4. Conclusão

Este artigo apresentou a OBI-UAN, uma ferramenta que combina modelos de linguagem LLMs, técnicas de recuperação de informações e uma interface web desenvolvida com Streamlit para apoiar estudantes na preparação para a OBI e para o mercado de trabalho. A proposta se destaca por oferecer funcionalidades inéditas, como categorização das questões por tema e nível de dificuldade, além de *feedback* automatizado para respostas, contribuindo diretamente para o processo de aprendizagem.

Como trabalhos futuros, pretendemos avaliar o uso de outros modelos de linguagem para aprimorar a qualidade dos *feedbacks* gerados, bem como incorporar novas formas de categorização, como tópicos específicos de algoritmos (ex.: busca, ordenação,

¹³<https://huggingface.co/>

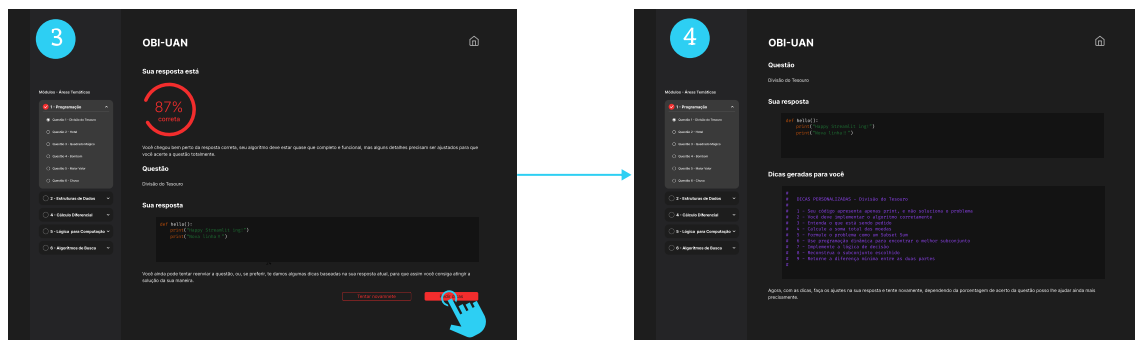


Figura 3. Tela de Resultados para a Tela de *Feedback*

programação dinâmica) ou classificação por habilidades cognitivas (ex.: análise, síntese, raciocínio lógico). Também planejamos melhorias na interface, visando facilitar a organização visual do plano de estudo e tornar a experiência do usuário mais intuitiva e eficiente.

Referências

- da Silva Pereira, G., Piaia, E., Nadalleti, L., Rossetto, N., and dos Santos Ádario, A. M. (2019). Obi-iniciação à lógica e à programação. In *8º JEPEX e 2º Mostra*.
- de Lima, J. R. and de Menezes, C. S. (2024). As dificuldades enfrentadas pelos estudantes na aprendizagem de programação de computadores: Uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação*, 22(1):130–140.
- MELO, D., MELO, D. C. A. d., et al. (2023). Analisando llms na resolução de problemas de competições de programação: um estudo com chatgpt e bard.
- Silva, T. L. d., Vidotto, K. N. S., Tarouco, L. M. R., and Silva, P. F. d. (2024). Inteligência artificial generativa no ensino de programação: um mapeamento sistemático da literatura. *RENTE: Novas Tecnologias na Educação. Porto Alegre, RS. Vol. 22, n. 1 (jul. 2024)*, p. 262-272.
- SOUZA, J. V. d. M. C. (2024). Avaliando modelos de linguagem grande na resolução de problemas de lógica da obi.
- Teixeira, A. R. and Alliprandini, P. M. Z. (2013). Intervenção no uso de estratégias de aprendizagem diante de dificuldades de aprendizagem. *Psicologia escolar e educacional*, 17:279–288.
- Wang, L., Shi, C., Du, S., Tao, Y., Shen, Y., Zheng, H., and Qiu, X. (2024). Performance review on llm for solving leetcode problems. In *2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing (AIIM)*, pages 1050–1054. IEEE.
- Yoshida, C., Matsushita, M., and Higo, Y. (2024). Estimating the difficulty of programming problems using fine-tuned llm. In *2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 28–34. IEEE.