

Geração de Dicas de Próximo Passo utilizando Large Language Models (LLMs)

Luis Gustavo Araujo¹, Ketlly Araujo¹, Endrik Reis¹, Mirele Oliveira da Silva¹, Andreia Pinheiro²

¹Instituto Federal de Educação, Ciência e Tecnologia da Bahia – IFBA Campus Jacobina

²Prefeitura Municipal de Juazeiro – Ba

luis_araujo@ifba.edu.br, [{ketllyaraujo79, endriksantos19, mirelee3456, andreiapinheiro.ifba@gmail.com}](mailto:{ketllyaraujo79, endriksantos19, mirelee3456, andreiapinheiro.ifba@gmail.com)

Com a popularização dos Grandes Modelos de Linguagem (*Large Language Models* – LLMs), seu uso se intensificou e os domínios de aplicação se expandiram [Denny et al. 2024]. Dentre esses domínios, destacam-se os assistentes de erro, que fornecem feedback para auxiliar os usuários na realização correta de uma tarefa, como o Copilot [Wermelinger 2023] e outros. Entre os tipos de feedback que podem ser fornecidos, é possível solicitar aos modelos mais informações sobre o problema, detalhes sobre o erro atual e como resolvê-lo ou dicas de como prosseguir, também conhecidas como dicas de próximo passo. Estas dicas são particularmente relevantes em contextos como o ensino de programação, no qual os estudantes realizam atividades suscetível à segmentação em passos. Diversos autores têm trabalho na avaliação da eficácia dessas dicas [Roest et al. 2024, Birillo et al. 2024, Agrawal et al. 2024, Roest 2023]. Em grande parte, essas dicas são fornecidas pelos professores para auxiliar os alunos. No entanto, em turmas numerosas, o feedback individualizado pode se tornar inviável. Além disso, muitos estudantes não conseguem receber retorno dos professores em tempo hábil. Diante desse cenário, este trabalho propõe a construção de um ambiente virtual de aprendizagem que integra um editor de código e um agente emissor de dicas de próximo passo utilizando LLMs. Além de login e lista de atividades, o ambiente inclui um editor de código. O editor conta com um espaço para codificação, um console para entrada e saída de dados e botões para exibir a descrição da atividade, exibir a próxima dica e executar o código. O sistema foi projetado para ser executado em navegadores. Utilizamos o Skulpt como interpretador Python, além do modelo GPT-4.0 para geração de dicas. As tecnologias empregadas foram HTML, CSS, JavaScript, PHP e MySQL. A descrição das atividades e os passos necessários para sua conclusão são armazenados no banco de dados e consultados no momento da requisição via API da OpenAI. Para a formulação dos prompts, utilizamos a estratégia *few-shot*, fornecendo ao modelo alguns exemplos de perguntas e respostas a serem seguidos. Além desses exemplos, o prompt inclui a descrição da atividade, a lista de passos necessários para sua conclusão, o código atual do estudante e uma solicitação. Ao acessar a plataforma, o estudante escolhe uma atividade para realizar. Todos os botões referentes às atividades direcionam para o editor, onde é exibida a descrição da atividade. O principal objetivo dessa ferramenta é que o modelo forneça, de forma clara, a próxima ação que o estudante deve realizar com base no seu estado atual. Como trabalhos futuros, pretendemos desenvolver uma interface para o cadastro de questões pelos professores, um sistema de geração automática da lista de passos necessários para a conclusão da atividade utilizando LLMs e um mecanismo de validação desses passos pelos docentes.

Referências

- Agrawal, V., Singla, P., Miglani, A. S., Garg, S., and Mangal, A. (2024). Give me a hint: Can llms take a hint to solve math problems? *arXiv preprint arXiv:2410.05915*.
- Birillo, A., Artser, E., Potriasaeva, A., Vlasov, I., Dzialets, K., Golubev, Y., Gerasimov, I., Keuning, H., and Bryksin, T. (2024). One step at a time: Combining llms and static analysis to generate next-step hints for programming tasks. In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, pages 1–12.
- Denny, P., Gulwani, S., Heffernan, N. T., Käser, T., Moore, S., Rafferty, A. N., and Singla, A. (2024). Generative ai for education (gaied): Advances, opportunities, and challenges. *arXiv preprint arXiv:2402.01580*.
- Roest, L. (2023). Automated next-step hint generation for introductory programming using large language models. Master's thesis.
- Roest, L., Keuning, H., and Jeuring, J. (2024). Next-step hint generation for introductory programming using large language models. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 144–153.
- Wermelinger, M. (2023). Using github copilot to solve simple programming problems. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 172–178.