Análise dos efeitos do idioma na geração automática de respostas por aplicações de LLM

Fernando Roberto Delbone, Igor Scaliante Wiese, Marco Aurélio Graciotto Silva
fernandodelbone@alunos.utfpr.edu.br,{igor,magsilva}@utfpr.edu.br
Universidade Tecnológica Federal do Paraná – Departamento Acadêmico de Computação
Campo Mourão, Paraná, Brasil

O advento da técnica de *Large Language Model* (LLM) trouxe oportunidades e desafios para a comunidade de educação em Computação [1, 4, 6, 8]. Atualmente os modelos conseguem resolver com sucesso problemas tipicamente utilizados em disciplinas introdutórias de Computação (CS1) [3–5, 7, 9, 10]. No entanto, de modo geral os estudos consideraram problemas e *prompts* especificados em inglês. Assim, é de interesse conhecer o desempenho de modelos LLM para o contexto brasileiro, com problemas em português.

O objetivo deste trabalho é avaliar os efeitos do idioma utilizado na especificação do problema para a geração automática de respostas em problemas aplicáveis ao contexto de disciplinas de introdução à programação (CS1). Serão considerados problemas originalmente em português que foram traduzidos para inglês de forma automática e de forma manual. Dessa forma, a principal questão de pesquisa será: A qualidade da solução gerada automaticamente independe do idioma original do problema?

Para tratar a questão de pesquisa, foram consideradas as seguintes perspectivas. Primeiramente, ao utilizar um problema originalmente definido em português, as soluções geradas pelas aplicações LLM são tão boas quanto àquelas geradas para problemas originalmente definidos em inglês? A segunda perspectiva considera problemas originalmente definidos em português, mas traduzidos automaticamente para inglês. Finalmente, a terceira perspectiva considera problemas traduzidos para inglês por um especialista.

Para cada perspectiva, foram consideradas aplicações LLM para a geração das respostas. Devido à maioria dos artigos acadêmicos abordarem o ChatGPT (modelos GPT 3.0, 3.5 e 4.0), ele foi incluído na avaliação (modelo GPT 3.5). Além dele, também foram considerados o Google Bard (modelo PaLM 2) e o HuggingFace HuggingChat (modelo Mixtral-8x7B). Dessa forma, espera-se entender o impacto do idioma face aos modelos LLM empregados por cada aplicação.

Para cada problema a ser resolvido pelo estudante, foi definida uma estrutura para sua descrição, composta de enunciado, formatação do resultado a ser apresentado, dicas para resolução e casos de teste. Esses elementos, por sua vez, foram utilizados para construir o *prompt* de entrada para cada aplicação LLM a ser avaliada. Para avaliar o efeito dessa estrutura definida, também foram construídos quatro *prompts*, incluindo gradativamente tais elementos.

Os problemas foram obtidos de um repositório de atividades avaliadas automaticamente por um juiz online [2]. Atualmente foram considerados três problemas, respectivamente envolvendo variáveis, comandos de repetição e vetores. Considerando o conjunto de dados e publicações associadas, foram extraídos os enunciados e os casos de teste e, quando disponíveis ou manualmente, foram definidas a formatação dos resultados e dicas de resolução.

Dos casos de teste definidos, alguns poucos foram utilizados para a construção dos *prompt* e os demais foram empregados para avaliar a correção das soluções geradas pelas aplicações LLM, de forma similar às aplicações de avaliação automática utilizadas em educação em Computação. Assim, o critério de avaliação foi a quantidade de casos de teste satisfeitos pelo código, ou seja, quanto maior essa quantidade, maior a qualidade do código.

Conforme descrito no método, foram especificados os problemas e construídos *prompts* e aplicados no ChatGPT, Google Bard e HuggingChat. Por enquanto foram definidos três problemas e respectivos *prompts*. Para cada um deles, foi obtido o código da resolução pelas aplicações LLM, avaliando-se a correção pela proporção de casos de teste corretos. Os dados completos estão disponíveis em https://github.com/FerDelbo/Repositorio_IC.

Um dos problemas analisados tratava do cálculo da quantidade de veículos que infringiram a velocidade máxima de uma via, empregando conceitos de vetores. Para o problema em português, o ChatGPT acertou 75% dos casos de teste, seguido do Bard com 66% e o HuggingChat com 0%. Assim, observa-se uma taxa de acerto elevada para os dois primeiros e zerada para o último. Ao usar a versão traduzida automaticamente para inglês, os resultados foram 75%, 33% e 0%, respectivamente. Ainda está em andamento a realização do estudo com a tradução revisada. Desta forma, observa-se que a taxa de acerto para o ChatGPT e Bard são elevadas, tanto para problemas em português ou inglês, enquanto o HuggingChat não alcançou bons resultados. Ao usar a tradução automática, a qualidade da solução reduziu no caso do Bard.

Quanto à estrutura do *prompt*, nenhuma das aplicações LLM conseguiu acertar a resposta apenas com o enunciado. No entanto, ao fornecer a formatação, tanto o ChatGPT quanto o Bard forneceram respostas corretas, enquanto o HuggingChat falha. Ao acrescentar a dica de resolução, o Bard oferece uma resposta errada e o HuggingChat continua falhando. Ao acrescentar os casos de teste, o Bard volta a acertar, mas o HuggingChat ainda falha. Assim, observa-se que o acréscimo de elementos na estrutura geralmente permite um desempenho melhor das aplicações.

Assim, espera-se, com este trabalho, contribuir com a educação em Computação ao mensurar os efeitos do idioma das atividades ao utilizar aplicações LLM para geração automática de respostas, evitando respostas incorretas a estudantes e prejudicar seu aprendizado. Na perspectiva docente, a devida dimensão dos efeitos do idioma permitirá o uso mais eficaz das aplicações LLM no ensino.

EduComp24, Abril 22-27, 2024, São Paulo, São Paulo, Brasil (On-line)

AGRADECIMENTOS

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Processo 408812/2021-4 - e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

REFERÊNCIAS

[1] Brett A. Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2023. Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education* (54 ed.) (Toronto, ON, Canadá). ACM, New York, NY, EUA, 500–506.

[2] CodeBench. 2023. CodeBench Educational Mining Dataset 1.80. Conjunto de dados. https://codebench.icomp.ufam.edu.br/dataset/

[3] Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education* (54 ed.) (Toronto, ON, Canadá). ACM, New York, NY, EUA, 1136–1142.

[4] Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing Education in the Era of Generative AI. *Commun. ACM* 67, 2, 56–67.

[5] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *24th Australasian Computing Education Conference (ACE 2022)* (24 ed.) (Austrália), Judy Sheard and Paul Denny (Eds.). ACM, New York, NY, EUA, 10–19.

[6] Sam Lau and Philip Guo. 2023. From "Ban It Till We Understand It"to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools Such as ChatGPT and GitHub Copilot. In *ACM Conference on International Computing Education Research (ICER) 2023* (19 ed.) (Chicago, IL, EUA), Kathi Fisler and Paul Denny (Eds.). ACM, New York, NY, EUA, 106–121.

[7] Stephen R. Piccolo, Paul Denny, Andrew Luxton-Reilly, Samuel H. Payne, and Perry G. Ridge. 2023. Evaluating a large language model's ability to solve programming exercises from an introductory bioinformatics course. *PLOS Computational Biology* 19, 9, 1–16.

[8] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *2023 Conference on Innovation and Technology in Computer Science Education* (Turku, Finlândia). ACM, New York, NY, EUA, 108–159.

[9] Mike Richards, Kevin Waugh, Mark Slaymaker, Marian Petre, John Woodthorpe, and Daniel Gooch. 2024. Bob or Bot: Exploring ChatGPT's Answers to University Computer Science Assessment. *Transactions on Computing Education* 24, 1, 5:1–5:32.

[10] Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023. Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. In *ACM Conference on International Computing Education Research (ICER) 2023* (19 ed.) (Chicago, IL, EUA), Kathi Fisler and Paul Denny (Eds.). ACM, New York, NY, EUA, 78–92.