Multimodal Analogy Generation in Programming Education

Noviello, Yuri; Birillo, Anastasiia; Migut, Gosia

Citation (APA)
Noviello, Y., Birillo, A., & Migut, G. (2025). Multimodal Analogy Generation in Programming Education. In *ITiCSE 2025 - Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education* (pp. 783). (Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE; Vol. 2). ACM. https://doi.org/10.1145/3724389.3730781

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Multimodal Analogy Generation in Programming Education

Yuri Noviello
Delft University of Technology
Delft, The Netherlands
y.noviello@tudelft.nl

Anastasiia Birillo
JetBrains Research
Belgrade, Serbia
anastasia.birillo@jetbrains.com

Gosia Migut
Delft University of Technology
Delft, The Netherlands
m.a.migut@tudelft.nl

Abstract

Engaging students with effective learning materials continues to be a significant challenge in programming education. Analogies are commonly used to simplify complex topics, enabling learners to relate unfamiliar concepts to familiar ones. Additionally, visual representations of these analogies can enhance engagement and improve the overall learning experience. This work presents a prototype of a novel AI tool that generates analogy-based explanations and corresponding video animations for programming education. The tool leverages Large Language Models (LLMs) for analogy generation and a structured animation workflow for visualization. This poster invites discussion on the effectiveness of AI-generated educational content and its implications for programming education.

CCS Concepts

• **Computing methodologies** → Artificial intelligence; • **Applied computing** → Education; • **Human-centered computing** → Visualization.

Keywords

Analogy Generation, Animation Generation, Programming Education, Artificial Intelligence, Multimodal Learning

1 Introduction

Educational programming material often struggles to make abstract concepts accessible. Analogies help simplify challenging topics by linking them to ideas familiar to the student [1, 4]. However, creating engaging, multimodal instructional materials remains resource-intensive, making it impractical for most educators [3]. Recent advancements in AI and NLP present an opportunity to automate the generation of analogy-driven explanations and corresponding visualizations. This research explores a multimodal AI-powered system that generates analogies for programming topics and transforms them into explanatory animations. We introduce a two-part tool: (1) **Analogy Generation Module** that uses LLMs to create textual analogies for programming concepts and (2) **Animation Generation Module**, that uses LLMs to generate `manim` code [5] to transform these analogies into dynamic, instructional animations. An example of the generated animation by the tool can be found in the Figure 1.
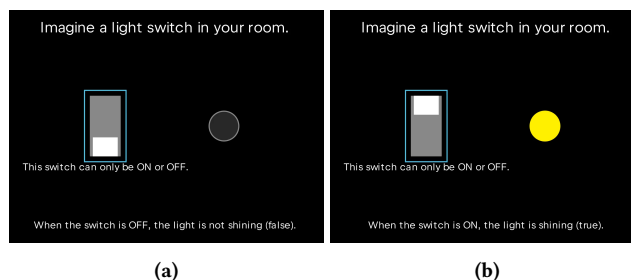
Figure 1: Frames extracted from LLM-generated animation explaining *Boolean* as a *Light Switch*.
(1a) The *Boolean* is `false`, meaning the light does not shine.
(1b) The *Boolean* is `true`, meaning the light shines.

2 Our Work

Our work explores the question: *How to effectively use AI to generate analogies and animations for programming education, when evaluated against established analogy-quality metrics, pedagogical value, and students satisfaction?*

This research focuses on three key aspects: (1) analogy generation, (2) animation creation, and (3) evaluation. We already have developed the prototype of the tool. To measure the effectiveness of AI-generated analogies and animations, we designed a multi-phase evaluation, involving both expert educators and students. Educators will assess the accuracy, clarity, and pedagogical alignment of the generated analogies [2] and animations. Students' feedback will be collected through surveys to measure engagement, usability, and cognitive load.

In future work, a comparative study will be performed to quantify learning outcomes by measuring students' comprehension and retention.

References

[1] Seth Bernstein, Paul Denny, Juho Leinonen, Lauren Kan, Arto Hellas, Matt Littlefield, Sami Sarsa, and Stephen Macneil. 2024. "Like a Nesting Doll": Analyzing Recursion Analogies Generated by CS Students Using Large Language Models. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (ITiCSE 2024). New York, NY, USA, (July 2024), 122–128. https://dl.acm.org/doi/10.1145/3649217.3653533.

[2] Bhavya Bhavya, Chris Palaguachi, Yang Zhou, Suma Bhat, and ChengXiang Zhai. 2024. Long-Form Analogy Evaluation Challenge. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*. Simon Mille and Miruna-Adriana Clinciu, (Eds.) Tokyo, Japan, (Sept. 2024), 1–16. https://aclanthology.org/2024.inlg-genchal.1/.

[3] Bettyjo Bouchey, Jill Castek, and John Thygeson. 2021. Multimodal Learning. en. In *Innovative Learning Environments in STEM Higher Education: Opportunities, Challenges, and Looking Forward*. Jungwoo Ryoo and Kurt Winkelmann, (Eds.) Cham, 35–54. https://doi.org/10.1007/978-3-030-58948-6_3.

[4] George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. Mark Johnson, (Ed.) Chicago.

[5] The Manim Community Developers. 2025. Manim – Mathematical Animation Framework. (Jan. 2025). https://www.manim.community/.