Using Large Language Models to Automatically Identify Programming Concepts in Code Snippets

Conference Paper · June 2023

DOI: 10.1145/3568812.3603482

Andrew Tran

Temple University

8 PUBLICATIONS **51** CITATIONS

SEE PROFILE

Egi Rama

Temple University

2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Kenneth Angelikas

Temple University

2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Stephen Macneil

Temple University

49 PUBLICATIONS **312** CITATIONS

SEE PROFILE

Using Large Language Models to Automatically Identify Programming Concepts in Code Snippets

Andrew Tran
Temple University
Philadelphia, PA, United States
andrew.tran10@temple.edu

Linxuan Li
Conestoga High School
Berwyn, US
linxuanliwork@gmail.com

Egi Rama
Temple University
Philadelphia, US
egi.rama@temple.edu

Kenneth Angelikas
Temple University
Philadelphia, US
kenneth.angelikas@temple.edu

Stephen MacNeil
Temple University
Philadelphia, PA, United States
stephen.macneil@temple.edu

ABSTRACT

Curating course material that aligns with students' learning goals is a challenging and time-consuming task that instructors undergo when preparing their curricula. For instance, it is a challenge to find multiple-choice questions or example codes that demonstrate recursion in an unlabeled question bank or repository. Recently, Large Language Models (LLMs) have demonstrated the capability to generate high-quality learning materials at scale. In this poster, we use LLMs to identify programming concepts found within code snippets, allowing instructors to quickly curate their course materials. We compare programming concepts generated by LLMs with concepts generated by experts to see the extent to which they agree. The agreement was calculated using Cohen's Kappa.

CCS CONCEPTS

• **Social and professional topics** → *Computing education*; • **Computing methodologies** → *Natural language generation.*

KEYWORDS

large language models, explanations, computer science education

ACM Reference Format:
Andrew Tran, Linxuan Li, Egi Rama, Kenneth Angelikas, and Stephen MacNeil. 2023. Using Large Language Models to Automatically Identify Programming Concepts in Code Snippets. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.2 (ICER '23 V2), August 7–11, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3568812.3603482

1 INTRODUCTION

Instructors face a sense-making challenge when trying to curate learning material for their courses [2]. To explore the potential for generating programming concepts found within code snippets, we expand on recent work using LLMs to generate learning material [1, 3, 6–9], such as programming assignments [1, 9] and explanations [3, 4, 6, 8].

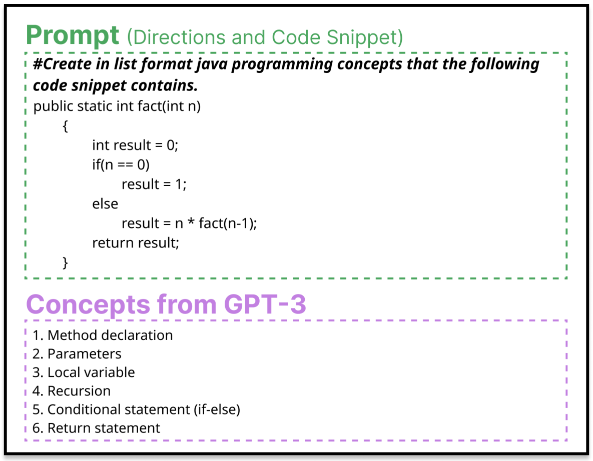
1.1 Methodology

We constructed a text corpus by randomly selecting 10 code examples from 10 Introduction to Java e-books, resulting in 100 code examples. We randomly sampled 50 of these examples from our text corpus for analysis. We then collect human-generated lists of programming concepts by having experts annotate the code examples. We report on the agreement between concepts identified by experts and the concepts generated by the LLM (GPT-3) using Cohen's Kappa Calculations (Figure 2).

2 DISCUSSION

2.1 Implications for the CS Classroom

Instructors may use LLMs to curate their course material and identify relevant course conceptsConcurrently, instructors would be able to identify missing material and close knowledge gaps.



#Create in list format java programming concepts that the following code snippet contains.

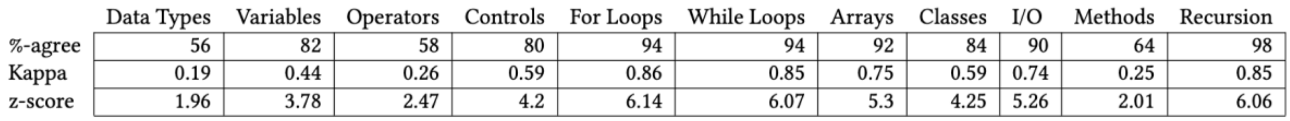
```
public static int fact(int n)
    {
        int result = 0;
        if(n == 0)
            result = 1;
        else
            result = n * fact(n-1);
        return result;
    }
```

1. Method declaration
2. Parameters
3. Local variable
4. Recursion
5. Conditional statement (if-else)
6. Return statement

Figure 1: Prompt, Code Snippet, and LLM-generated Concepts

ICER '23 V2, August 7–11, 2023, Chicago, IL, USA

ACM ISBN 978-1-4503-9975-3/23/08.
https://doi.org/10.1145/3568812.3603482



	Data Types	Variables	Operators	Controls	For Loops	While Loops	Arrays	Classes	I/O	Methods	Recursion
%-agree	56	82	58	80	94	94	92	84	90	64	98
Kappa	0.19	0.44	0.26	0.59	0.86	0.85	0.75	0.59	0.74	0.25	0.85
z-score	1.96	3.78	2.47	4.2	6.14	6.07	5.3	4.25	5.26	2.01	6.06

Figure 2: Results from Cohen's Kappa tests between LLM and Experts across 11 Introduction to Java concepts.

2.2 Implications for Students' Metacognition

Identifying programming concepts found in code may encourage students to think explicitly about the concepts they are learning and make connections between concepts and tangible examples. This explicit labeling may provide students with a *framework for monitoring* [5, 10] their learning.

3 CONCLUSION

Our poster presentation will include various examples of LLM-generated code snippets and concept lists. We will discuss cases where LLMs struggled to produce the correct concept. Future work will investigate the potential for automatically annotating code examples found in the wild. We hypothesize that repeated exposure to code examples labeled with relevant programming concepts may help students develop a familiarity with these concepts and a curiosity to learn more when encountering concepts they don't know. The potential value of these annotations, which may not have previously justified the effort, may become evident when implemented on a large scale.

REFERENCES

[1] Brett A. Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2022. Programming Is Hard – Or at Least It Used to Be: Educational Opportunities And Challenges of AI Code Generation.

[2] Heryan Ardhi Kusuma and Sigit Apriyanto. 2018. Strategy on Developing English Learning Material for Specific Purposes. *IJECA (International Journal of Education and Curriculum Application)* (2018).

[3] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing Code Explanations Created by Students and Large Language Models. *arXiv preprint arXiv:2304.03938* (2023).

[4] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1.* 563–569.

[5] Dastyni Loksa, Lauren Margulieux, Brett A. Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2022. Metacognition and Self-Regulation in Programming Education: Theories and Exemplars of Use. *ACM Trans. Comput. Educ.* 22, 4, Article 39 (sep 2022), 31 pages. https://doi.org/10.1145/3487050

[6] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 931–937. https://doi.org/10.1145/3545945.3569785

[7] Stephen MacNeil, Andrew Tran, Juho Leinonen, Paul Denny, Joanne Kim, Arto Hellas, Seth Bernstein, and Sami Sarsa. 2022. Automatically Generating CS Learning Materials with Large Language Models. *arXiv preprint arXiv:2212.05113* (2022).

[8] Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, and Ziheng Huang. 2022. Generating Diverse Code Explanations Using the GPT-3 Large Language Model. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 2* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, New York, NY, USA, 37–39. https://doi.org/10.1145/3501709.3544280

[9] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the 2022 ACM Conf. on International Computing Education Research - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, New York, NY, USA, 27–43. https://doi.org/10.1145/3501385.3543957

[10] Benjamin Xie, Dastyni Loksa, Greg L Nelson, Matthew J Davidson, Dongsheng Dong, Harrison Kwik, Alex Hui Tan, Leanne Hwa, Min Li, and Amy J Ko. 2019. A theory of instruction for introductory programming skills. *Computer Science Education* 29, 2-3 (2019), 205–253.