AI in Coding: How can code generation models support developing computational thinking skills?

Author: Rick Mulder
Supervisor: Xiaoling Zhang
Responsible Professor: Fenia Aivaloglou

1. Introduction

With the recent AI developments, code generation models are able to better support and create code for programming.
This leads to the question of how good the current state of AI code support is, and for what kind of programming activities it can be used.

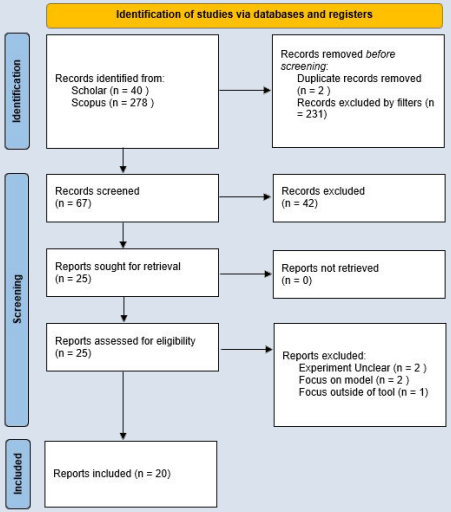
2. Research Question

RQ1: For what kind of programming support activities have the code generation models been used?
RQ2: How successful have they been considering these activities?

3. Method

We conducted a systematic literature review in order to find out how code generation models have been used in programming activities, either directly or by integrating it in a tool.
We synthesized the different works based on the supported activity.
The categories were extracted from the most commonly found activities, and a sixth category was made for other works that would fit in either none or multiple categories.

4. Results



PRISMA Flowchart

Activity	Findings
Code Generation	• Codex most used • Top 25% on introductory programming exercises • Sketch-based code generation promising results [2]
Code Documentation	• Inaccurate with Zero-shot learning • Few-shot learning allows adding extra details
Code Explanation	• Extensions in IDE's • 67% correct explanations • Explain error messages, and suggests correct fixes in 33% of the cases
Data Visualisation	• Natural language into SQL queries [1] • High success-rate with few-shot learning (80%) • Robust against under- and miss-specification
Software Vulnerability Detection	• Current Large Language Models perform well • High F1-score on most vulnerabilities (up to 93%) • Other vulnerabilities such as API function calls still harder to detect (78%) [5]
General Programming Support	• Co-coding with AI possibility • Support of non-programmers • More advanced IDE token prediction, like multi-token [5]

5. Conclusions

Several uses for GPT models to support programming have been found.
The current results are promising, but not yet applicable in a realistic environment.
However, at the current pace, it will not take long before we might see reliable AI support in programming.
The most important aspects for future research seem to be prompt engineering and finding the best ways to interface with these models.

6. References

[1] Trummer, Immanuel. **CodexDB: Generating Code for Processing SQL Queries using GPT-3 Codex** 2022.

[2] Li, Jia et al. **SkCoder: A Sketch-based Approach for Automatic Code Generation** 2023.

[3] Maddigan, Paula / Susnjak, Teo. **Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models** 2023.

[4] Chandra Thapa et al. **Transformer-based language models for software vulnerability detection.** page 481 – 496. Association for Computing Machinery, 2022.

[5] Maliheh Izadi et al. **Codefill: Multi-token code completion by jointly learning from structure and naming sequences.** volume 2022-May, page 401 – 412. IEEE Computer Society, 2022.