1. Zero-shot Prompting

- Usefulness: Very useful

- Justification: Enables developers to obtain quick responses or solutions without specific examples, aiding in handling new or unique coding challenges.

-always used this technique to understand training data.

2. Few-shot Prompting

- Usefulness: Very useful

- Justification: Allows developers to guide AI responses with a few examples, improving accuracy and context in code suggestions.

- used this technique after watching your YouTube vids to steer output for HW1

3. Chain-of-Thought Prompting

- Usefulness: Very useful

- Justification: Encourages step-by-step reasoning, which is valuable for complex problem-solving, debugging, and explaining code logic.

-haven't used yet for HW's but used for ML assignments

4. Self-Consistency

- Usefulness: Potentially useful after future AI development

- Justification: Could improve the reliability of AI responses, but current implementation may not be directly applicable to most software development tasks.

5. Generate Knowledge Prompting

- Usefulness: Very useful

- Justification: Helps gather information about unfamiliar technologies, algorithms, or best practices for application in coding tasks.

6. Prompt Chaining

- Usefulness: Very useful

- Justification: Breaks down complex tasks into smaller steps, beneficial for large-scale projects or intricate coding challenges.

-used for HW2

7. Tree of Thoughts

- Usefulness: Potentially useful after future AI development

- Justification: Promising for exploring multiple solution paths, but current implementation is too complex for day-to-day tasks.

8. Retrieval Augmented Generation

- Usefulness: Very useful

- Justification: Combines AI generation with retrieval from knowledge bases, enhancing accuracy and up-to-date code suggestions.

-

9. Automatic Reasoning and Tool-use

- Usefulness: Potentially useful after future AI development

- Justification: Promising for automating coding and testing, but current sophistication may not meet most development needs.

10. Automatic Prompt Engineer

- Usefulness: Less/not so useful

- Justification: Focuses on optimizing prompts rather than generating code, which is less directly applicable to programming tasks.

11. Active-Prompt

- Usefulness: Potentially useful after future AI development

- Justification: Could offer more interactive and adaptive coding assistance, but current implementation may not be broadly applicable.

12. Directional Stimulus Prompting

- Usefulness: Less/not so useful

- Justification: Interesting for some AI applications but typically requires more specific and structured inputs for software development.

13. Program-Aided Language Models

- Usefulness: Very useful

- Justification: Combines natural language processing with programmatic elements, leading to more accurate code generation and problem-solving.

14. ReAct

- Usefulness: Potentially useful after future AI development

- Justification: Interesting for combining reasoning and acting, but needs refinement for complex, multi-step coding challenges.

15. Reflexion

- Usefulness: Potentially useful after future AI development

- Justification: Self-reflection capabilities are intriguing, but immediate applicability to software tasks is limited.

16. Multimodal CoT

- Usefulness: Less/not so useful

- Justification: Powerful for certain AI tasks, but most software development focuses on text-based code rather than multimodal inputs.

17. Graph Prompting

- Usefulness: Potentially useful after future AI development

- Justification: Could handle complex, interconnected coding problems or system architectures, but needs further development.

Abhishek Ashish