

Self-Refine Prompting

Instructor

Bhaskarjit Sarmah

Vice President, Blackrock



Definition

- **Self-Refine Prompting** is an iterative process where a language model autonomously generates, assesses, and refines answers.
-

How Self-Refine Prompting Works?



Initiation

Begins with the model creating an initial response to a query.



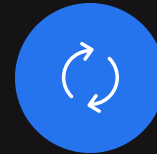
Review

The model evaluates its response, suggesting improvements.



Refinement

Adjustments are made to enhance the initial response.



Iteration

This cycle repeats until a satisfactory answer is not formed or a set attempt limit is reached.

Self-Refine Prompting: Example

Dialogue: x, y_t

User: I'm interested in playing table tennis.
Response: I'm sure it's a great way to socialize, stay active.

Feedback fb

Engaging: Provides no information about table tennis or how to play it.

User understanding: Lacks understanding of user's needs and state of mind.

REFINE

Response (refined): That's great to hear (...)! It's a fun sport requiring quick reflexes and good hand-eye coordination. Have you played before, or are you looking to learn?



The LLM refines dialogue by initially generating a basic response, then embracing it with feedback for greater user engagement and relevance.

Code optimization: x, y_t

```
Generate sum of 1, ..., N
def sum (n):
    res = 0
    for i in the range(n+1):
        res += i
    return res
```

Feedback fb

This code is slow as it uses brute force. A better approach is to use the formula $(n(n+1))/2$.

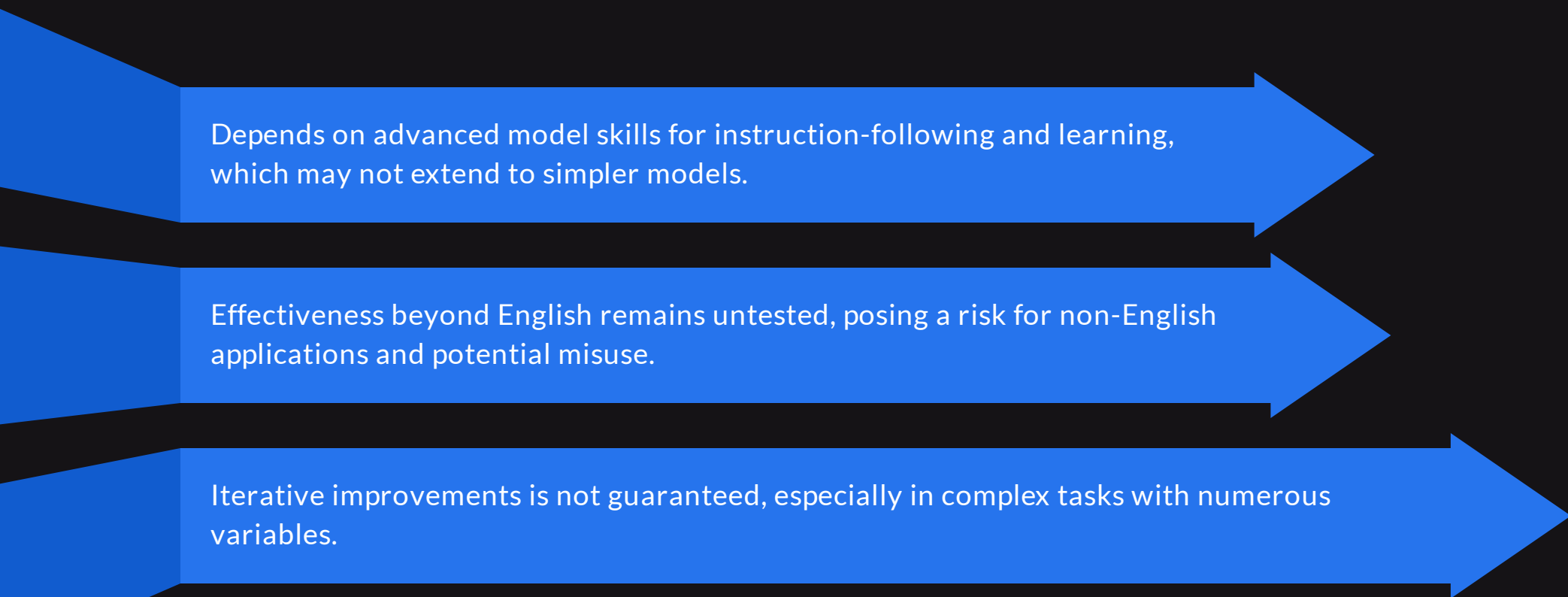
REFINE y_{t+1}

```
Code (refined)
def sum_faster (n):
    return (n*(n+1))/2
```



For code, the LLM starts with a simple function and then revises it using feedback to create a more efficient version.

Cons: Self-Refine Prompting



Depends on advanced model skills for instruction-following and learning, which may not extend to simpler models.

Effectiveness beyond English remains untested, posing a risk for non-English applications and potential misuse.

Iterative improvements is not guaranteed, especially in complex tasks with numerous variables.

Thank You
