

Rephrase and Respond Prompting

Instructor

Bhaskarjit Sarmah

Vice President, BlackRock



Definition

- Rephrase and Respond Prompting (RaR) is a method that adjusts query phrasing for clarity, enabling better understanding.
-

How Rephrase and Respond (RaR) Works?



One-step / two-step System

Rephrase, expand and respond



Make an impact

Combines with CoT for superior question-answering performance.

Rephrase and Respond Prompting: Example



"Take the last letters of the words in 'Edgar Bob' and concatenate them."

Rephrase and expand the question, and respond



Could you please form a new string or series of characters by joining together the final letters from each word in the phrase "Edgar Bob"?

The last letters in the words "Edgar" and "Bob" are "r" and "b", hence when concatenated, it forms "rb".



"Take the last letters of the words in 'Edgar Bob' and concatenate them."

Given the above question, rephrase and expand it to help you do better answering.



(Original) Take the last letters of the words in "Edgar Bob" and concatenate them.

Can you identify and extract the final letters in both the words that form 'Edgar Bob' and then join them together in the order they appear?

Rephrasing LLM



Can you identify and extract the final letters in both the words that form 'Edgar Bob' and then join them together in the order they appear?

Responding LLM



The last letters in the words "Edgar Bob" are "r" and "b". Concatenating them in the order they appear would be "rb".

Cons: Rephrase and Respond Prompting

RaR can enhance LLM's responses but may falter with complex tasks like deciphering first character in a Chinese idiom.

Poor training examples can lead LLMs to incorrect answers.

Without guidance (Zero-shot CoT), LLMs might generate nonsensical or biased steps.

Training questions require careful construction for clarity and accuracy.

Thank You

Self-Refine Prompting

Instructor

Bhaskarjit Sarmah

Vice President, Blackrock



Chain-of-verification

Instructor

Bhaskarjit Sarmah

Vice President, Blackrock



Definition

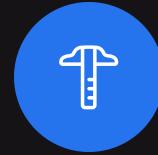
- Chain-of-Verification enables large language models to self-check and refine their responses through a series of validation questions.
- It includes a 'factored' approach to improve response accuracy by preventing the propagation of initial errors.

How Chain-of-Verification Works



Initial Responses

The model first drafts a provisional answer to the problem at hand.



Verification Questions

It then poses and addresses verification questions related to the initial answer.



Final Responses

Using insights from the verification step, the model produces a more accurate and validated final response.

Chain-of-Verification: Example

Query Name some politicians who were born in NY, New York.

Baseline Response

Here are some politicians who were born in NY, New York:

1. Hillary Clinton - former secretary of state and former Democratic presidential nominee
 2. Donald Trump - former president of the United States
 3. Michael Bloomberg - former Mayor of New York city and former Democratic presidential candidate
- <list continues...>

Plan Verifications

- Where was Hillary Clinton born?
- Where was Donald Trump born?
- Where was Michael Bloomberg born?

Execute Verifications

Hillary Clinton was born in Chicago, Illinois, USA on October 26, 1947.
Donald Trump was born on June 14, 1946, in Queens, NY, USA
Michael Bloomberg was born on Feb 14, 1942, in Boston, Massachusetts, USA

Final Verified Response

Here are some politicians who were born in NY, New York:

1. Donald Trump - former president of the USA
2. Alexandra Ocasio-Cortez -Democratic member of the U.S. House of Representatives.

Cons: Chain-of-Verifications

It reduce but doesn't entirely prevent incorrect model-generated content.

Tackles factual errors more than subtle reasoning or opinion inaccuracies.

Adds clarity but at a higher computational cost due to increased output length.

Its effectiveness is constrained by the model's inherent capabilities, such as identifying and knowing what it knows.

Thank You

Chain of Natural Language Inferencing

Instructor

Bhaskarjit Sarmah

Vice President, Blackrock



Definition

- **Chain of Natural Language Inferencing** is a hierarchical framework designed to address and reduce hallucinations in text generated by large language models (LLMs).
- Provides a structured approach to enhance the reliability of model-generated context without needing fine-tuning or specific prompts.

How Chain of Natural Language Inferencing Works



Detection Phase

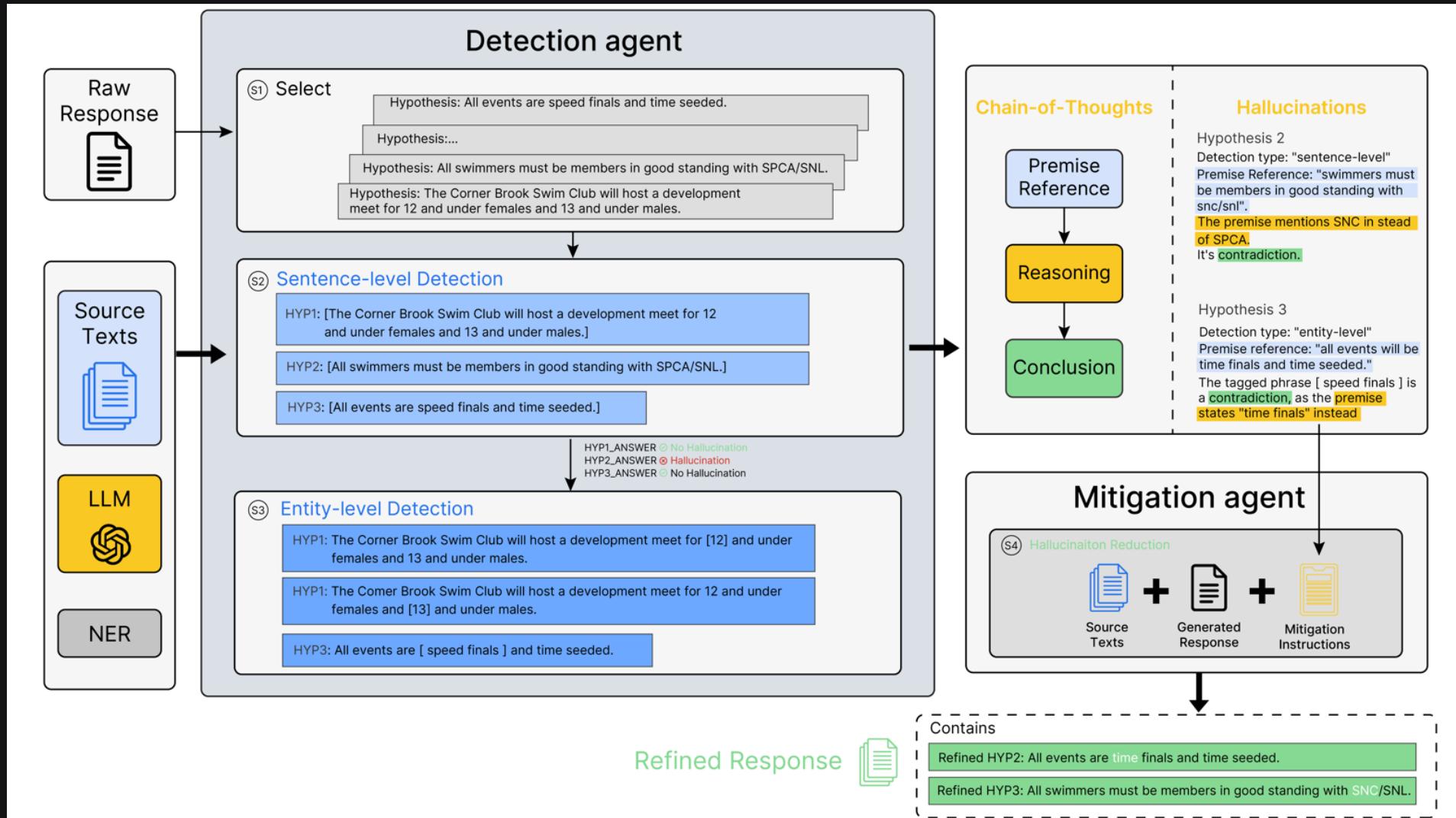
Employs a detection agent to identify hallucinations in the text.



Mitigation Phase

Uses a mitigation agent to refine or remove detected inaccuracies, preserving text fluency and coherence.

Chain of Natural Language Inferencing: Example



Cons: Chain of Natural Language Inferencing

May miss some hallucinations, mainly detecting those that lack grounding.

Effectiveness is dependent on the LLM's accuracy, which is not foolproof.

Post-processes rather than preventing hallucinations at the source.

Less effective for brief or fact-sparse responses due to its segmentation reliance.

Thank You

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.



Refinement

Adjustments are made to enhance the initial response.



Review

The model evaluates its response, suggesting improvements.



Iteration

This cycle repeats until a satisfactory answer is 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
