Prompting LLMs Using Haystack

Experiment with PromptNode and new prompt templates

Haystack Webinar

Vladimir Blagojevic

April 4th, 2023

Agenda

1. Introduction
2. Prompt Engineering - Fundamentals & Guidelines
3. Common Prompting Pitfalls
4. Latest Research
5. Q&A

01

Introduction

LLMs and their Impact on Our Society

- “The hottest new programming language is English” - Andrej Karpathy
- LLMs represent a world model
- Innovative application and transformative potential
- Prompting is a critical skill to harness LLMs
- Haystack LLMs API abstractions: `PromptNode`, `PromptTemplate`

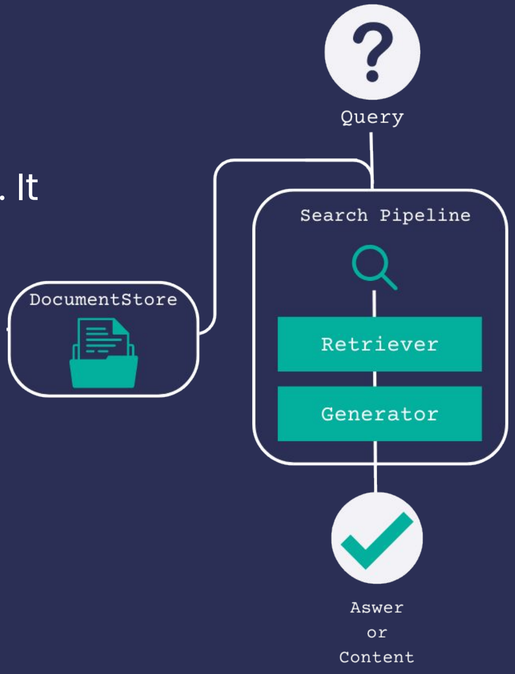
Recap: What’s available in Haystack?

`PromptNode`

A node that allows you to interact with LLMs in a customized way. It allows you to use models from OpenAI and Hugging Face.

`PromptTemplate`

A component that allows you to build flexible prompts that can be modified per query. This component can be provided To the `PromptNode` as a blueprint of how to interact with LLMs



02

Prompt Engineering – Fundamentals & Guidelines

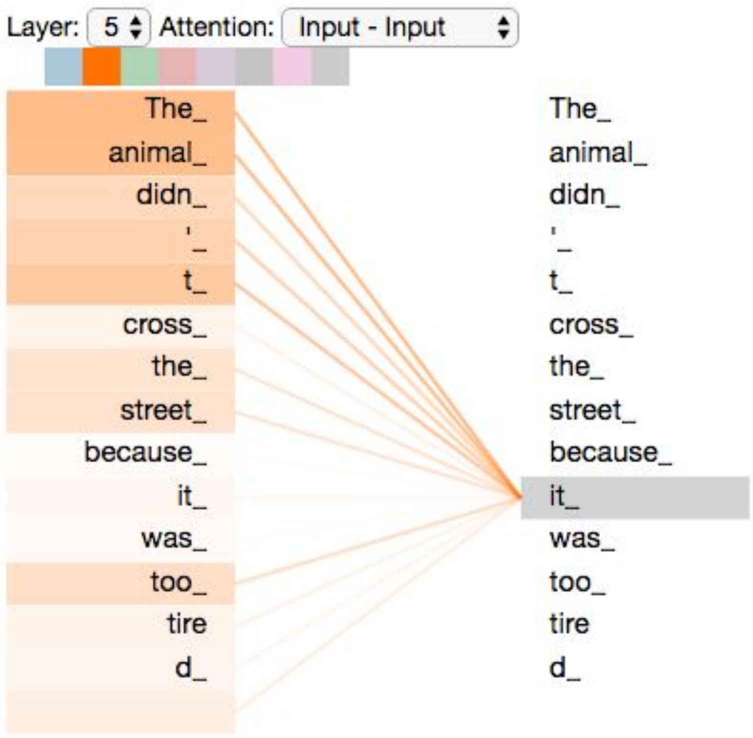
Fundamentals of Prompt Engineering

Why fundamentals?

Context window limitations

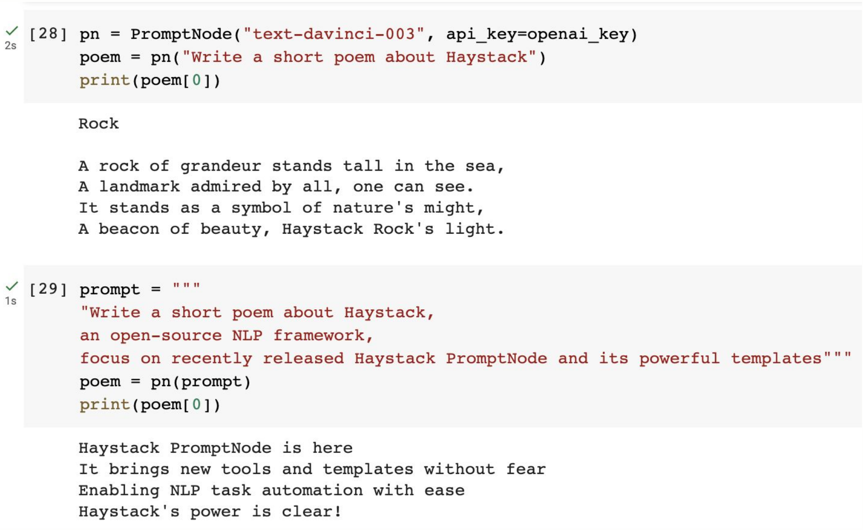
Attention mechanism

Model name	Window
flan-t5	512
text-davinci-003	4,097
gpt-4	8,192 32,768



Layer: 5 Attention: Input - Input

Fundamentals of Prompt Engineering



```
✓ [28] pn = PromptNode("text-davinci-003", api_key=openai_key)
2s     poem = pn("Write a short poem about Haystack")
       print(poem[0])
```

```
Rock

A rock of grandeur stands tall in the sea,
A landmark admired by all, one can see.
It stands as a symbol of nature's might,
A beacon of beauty, Haystack Rock's light.
```

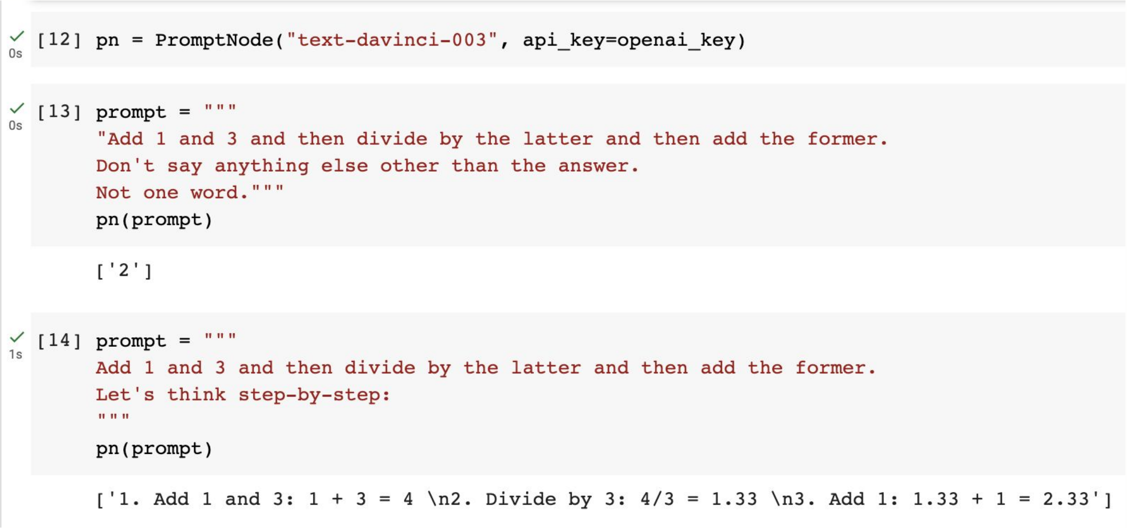
```
✓ [29] prompt = """
1s     "Write a short poem about Haystack,
       an open-source NLP framework,
       focus on recently released Haystack PromptNode and its powerful templates"""
       poem = pn(prompt)
       print(poem[0])
```

```
Haystack PromptNode is here
It brings new tools and templates without fear
Enabling NLP task automation with ease
Haystack's power is clear!
```

Clarity and specificity

Fundamentals of Prompt Engineering

Incorporating leading words (e.g. “let’s think step by step”)



```
[12] pn = PromptNode("text-davinci-003", api_key=openai_key)
```

```
[13] prompt = """
     "Add 1 and 3 and then divide by the latter and then add the former.
     Don't say anything else other than the answer.
     Not one word."""
     pn(prompt)
```

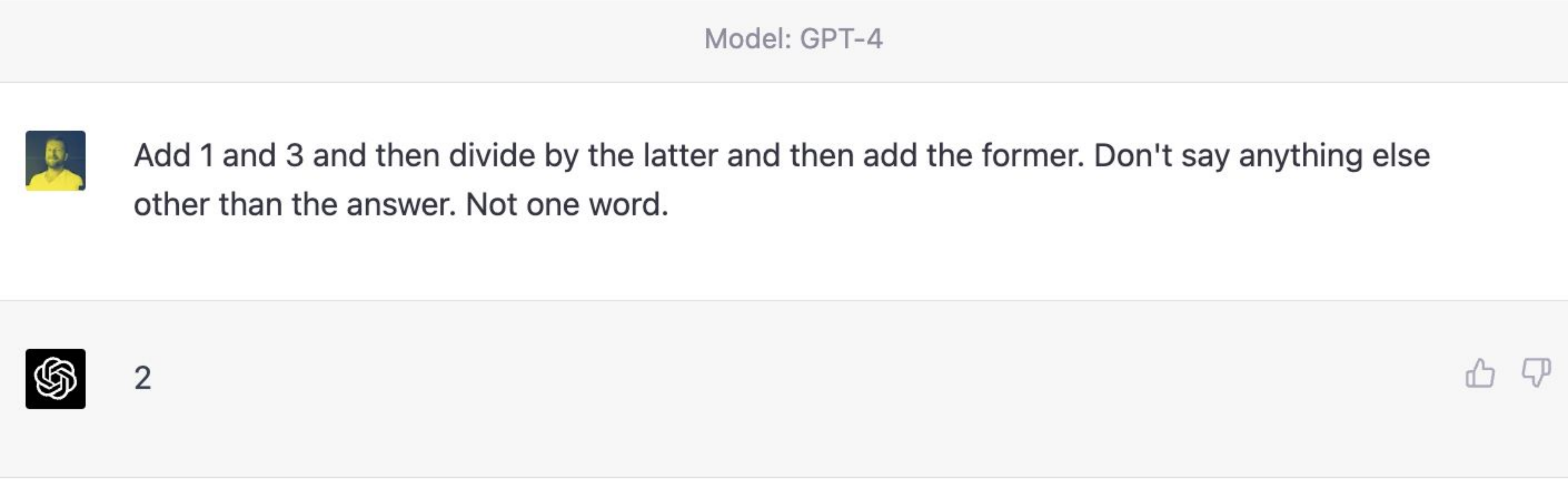
```
     ['2']
```

```
[14] prompt = """
     Add 1 and 3 and then divide by the latter and then add the former.
     Let's think step-by-step:
     """
     pn(prompt)
```

```
     ['1. Add 1 and 3: 1 + 3 = 4 \n2. Divide by 3: 4/3 = 1.33 \n3. Add 1: 1.33 + 1 = 2.33']
```

Fundamentals of Prompt Engineering

Yeah, ok, but GPT-4 doesn’t need leading words, right?!

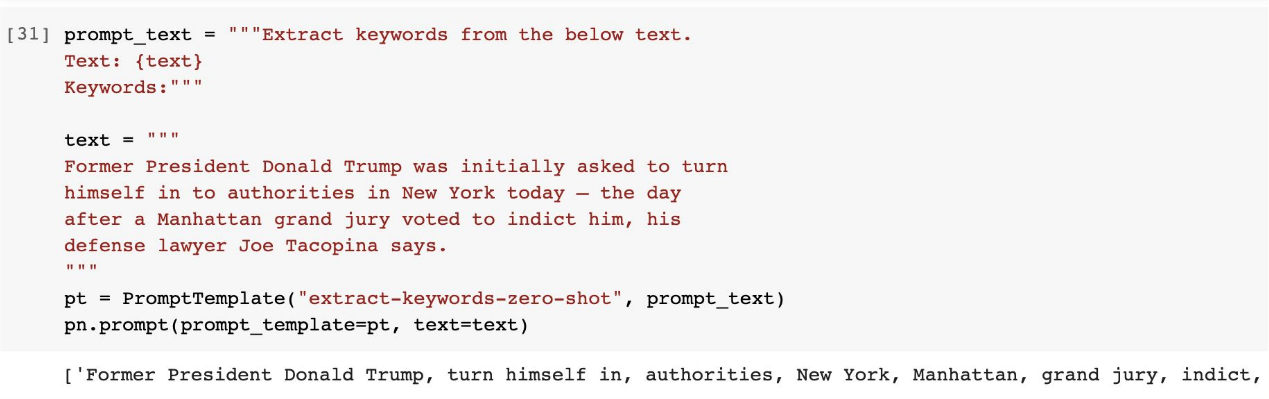


Model: GPT-4

Add 1 and 3 and then divide by the latter and then add the former. Don't say anything else other than the answer. Not one word.

2

Fundamentals of Prompt Engineering

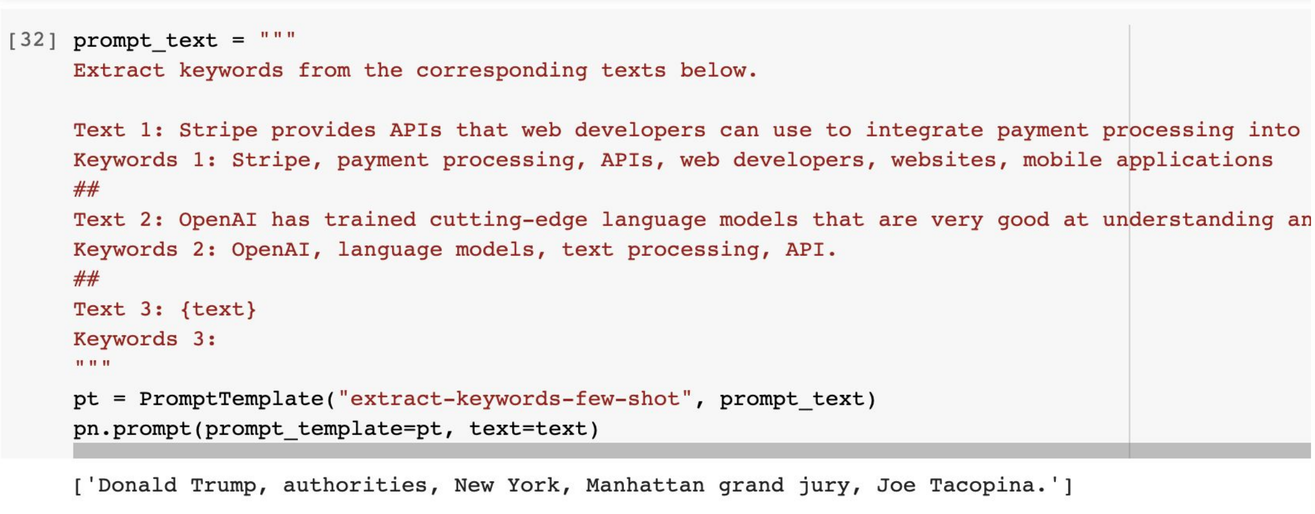


```
[31] prompt_text = """Extract keywords from the below text.
     Text: {text}
     Keywords:"""

     text = """
     Former President Donald Trump was initially asked to turn
     himself in to authorities in New York today — the day
     after a Manhattan grand jury voted to indict him, his
     defense lawyer Joe Tacopina says.
     """
     pt = PromptTemplate("extract-keywords-zero-shot", prompt_text)
     pn.prompt(prompt_template=pt, text=text)

     ['Former President Donald Trump, turn himself in, authorities, New York, Manhattan, grand jury, indict,
```

Start zero-shot

Fundamentals of Prompt Engineering

Continue with few-shot



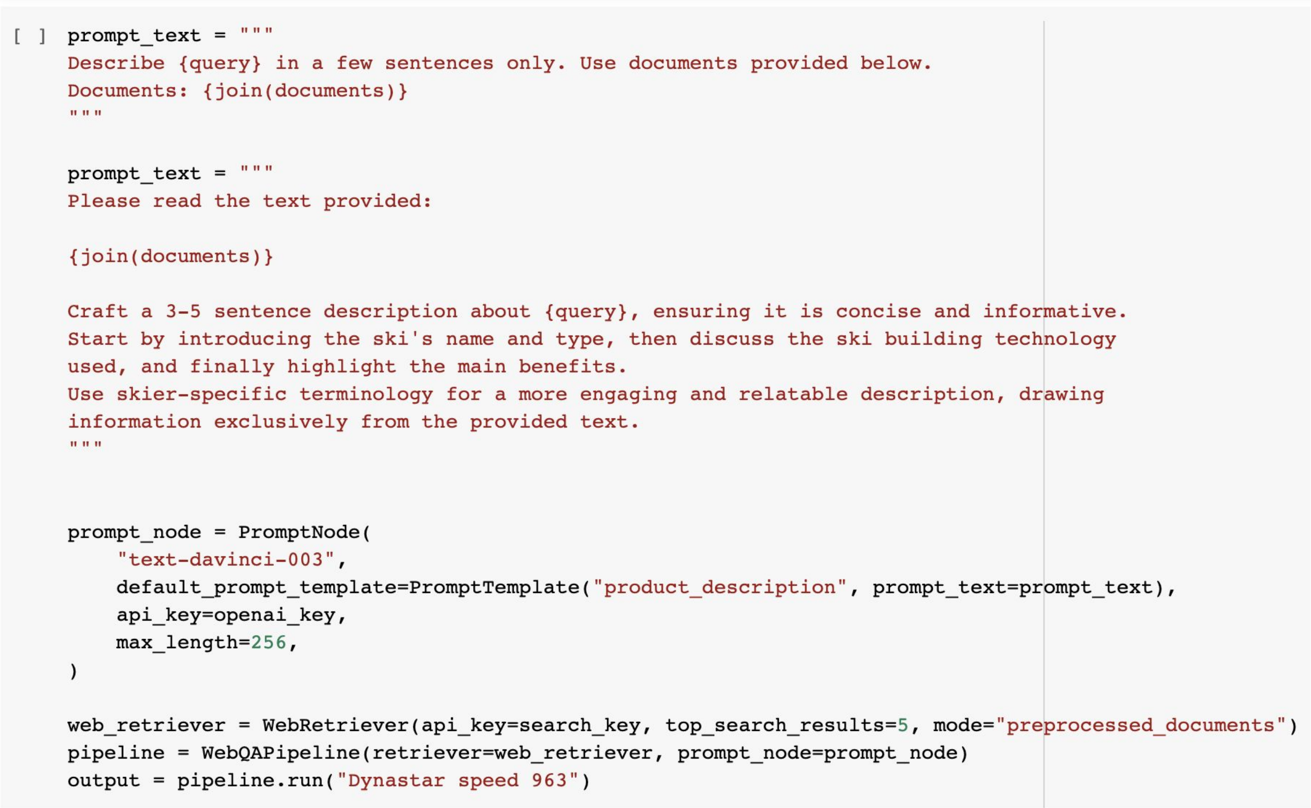
```
[32] prompt_text = """
     Extract keywords from the corresponding texts below.

     Text 1: Stripe provides APIs that web developers can use to integrate payment processing into
     Keywords 1: Stripe, payment processing, APIs, web developers, websites, mobile applications
     ##
     Text 2: OpenAI has trained cutting-edge language models that are very good at understanding an
     Keywords 2: OpenAI, language models, text processing, API.
     ##
     Text 3: {text}
     Keywords 3:
     """
     pt = PromptTemplate("extract-keywords-few-shot", prompt_text)
     pn.prompt(prompt_template=pt, text=text)
```

```
['Donald Trump, authorities, New York, Manhattan grand jury, Joe Tacopina.']
```

Prompt Engineering Guidelines

- Provide clear and explicit instructions
- Include examples if needed
- Experiment with different prompt styles
- Utilize system-level constraints (temperature, token limits)
- Iterate and refine prompts based on feedback
- Address potential biases in LLMs and prompts

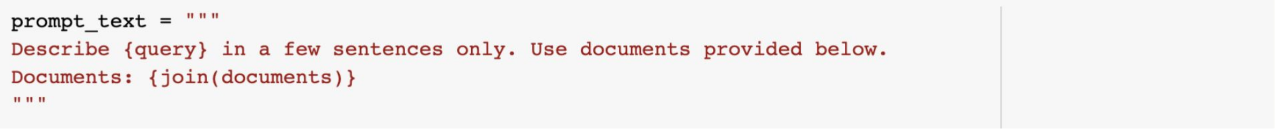
Prompt Engineering Guidelines



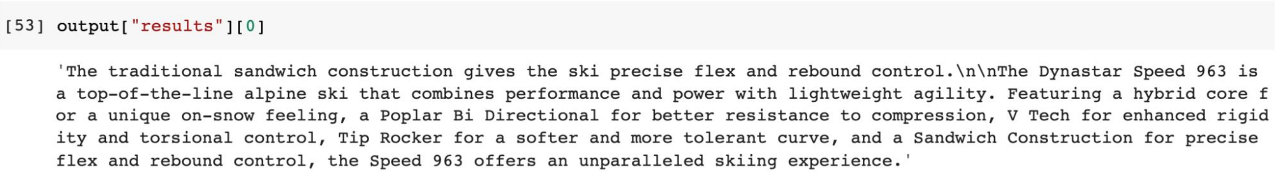
```
prompt_text = """
Describe {query} in a few sentences only. Use documents provided below.
Documents: {join(documents)}
"""

prompt_text = """
Please read the text provided:

{join(documents)}

Craft a 3-5 sentence description about {query}, ensuring it is concise and informative.
Start by introducing the ski's name and type, then discuss the ski building technology
used, and finally highlight the main benefits.
Use skier-specific terminology for a more engaging and relatable description, drawing
information exclusively from the provided text.
"""


prompt_node = PromptNode(
    "text-davinci-003",
    default_prompt_template=PromptTemplate("product_description", prompt_text=prompt_text),
    api_key=openai_key,
    max_length=256,
)

web_retriever = WebRetriever(api_key=search_key, top_search_results=5, mode="preprocessed_documents")
pipeline = WebQAPipeline(retriever=web_retriever, prompt_node=prompt_node)
output = pipeline.run("Dynastar speed 963")
```

Summary of everything we learned

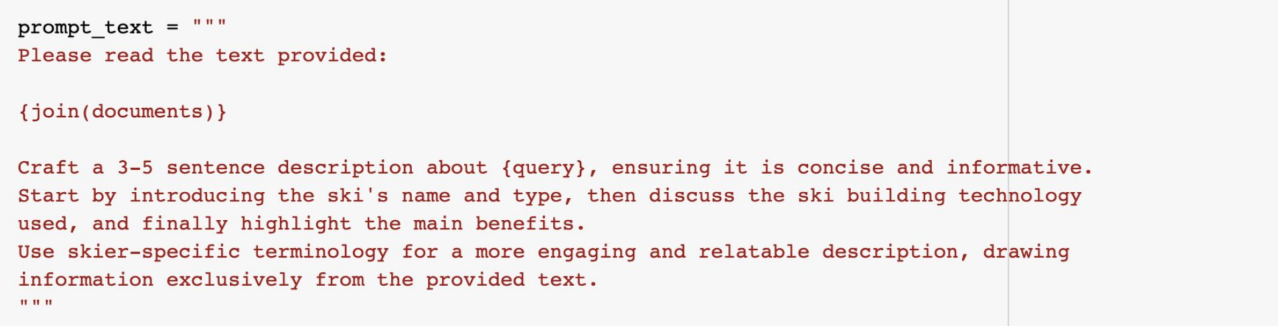
Prompt Engineering Guidelines



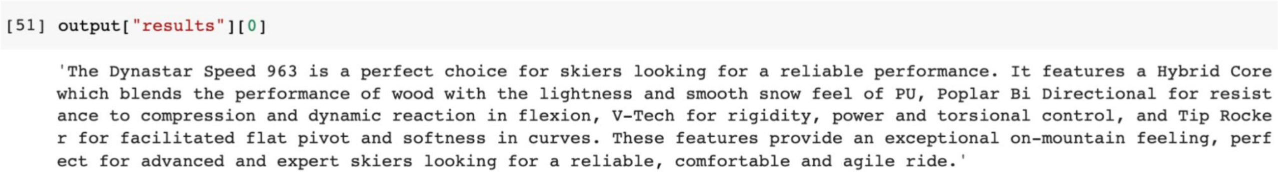
```
prompt_text = """
Describe {query} in a few sentences only. Use documents provided below.
Documents: {join(documents)}
"""
```



```
[53] output["results"][0]

    'The traditional sandwich construction gives the ski precise flex and rebound control.\n\nThe Dynastar Speed 963 is
    a top-of-the-line alpine ski that combines performance and power with lightweight agility. Featuring a hybrid core f
    or a unique on-snow feeling, a Poplar Bi Directional for better resistance to compression, V Tech for enhanced rigid
    ity and torsional control, Tip Rocker for a softer and more tolerant curve, and a Sandwich Construction for precise
    flex and rebound control, the Speed 963 offers an unparalleled skiing experience.'
```



```
prompt_text = """
Please read the text provided:

{join(documents)}

Craft a 3-5 sentence description about {query}, ensuring it is concise and informative.
Start by introducing the ski's name and type, then discuss the ski building technology
used, and finally highlight the main benefits.
Use skier-specific terminology for a more engaging and relatable description, drawing
information exclusively from the provided text.
"""
```



```
[51] output["results"][0]

    'The Dynastar Speed 963 is a perfect choice for skiers looking for a reliable performance. It features a Hybrid Core
    which blends the performance of wood with the lightness and smooth snow feel of PU, Poplar Bi Directional for resist
    ance to compression and dynamic reaction in flexion, V-Tech for rigidity, power and torsional control, and Tip Rocke
    r for facilitated flat pivot and softness in curves. These features provide an exceptional on-mountain feeling, perf
    ect for advanced and expert skiers looking for a reliable, comfortable and agile ride.'
```

Summary of everything we learned

03

Common Prompting Pitfalls

Common Prompting Pitfalls

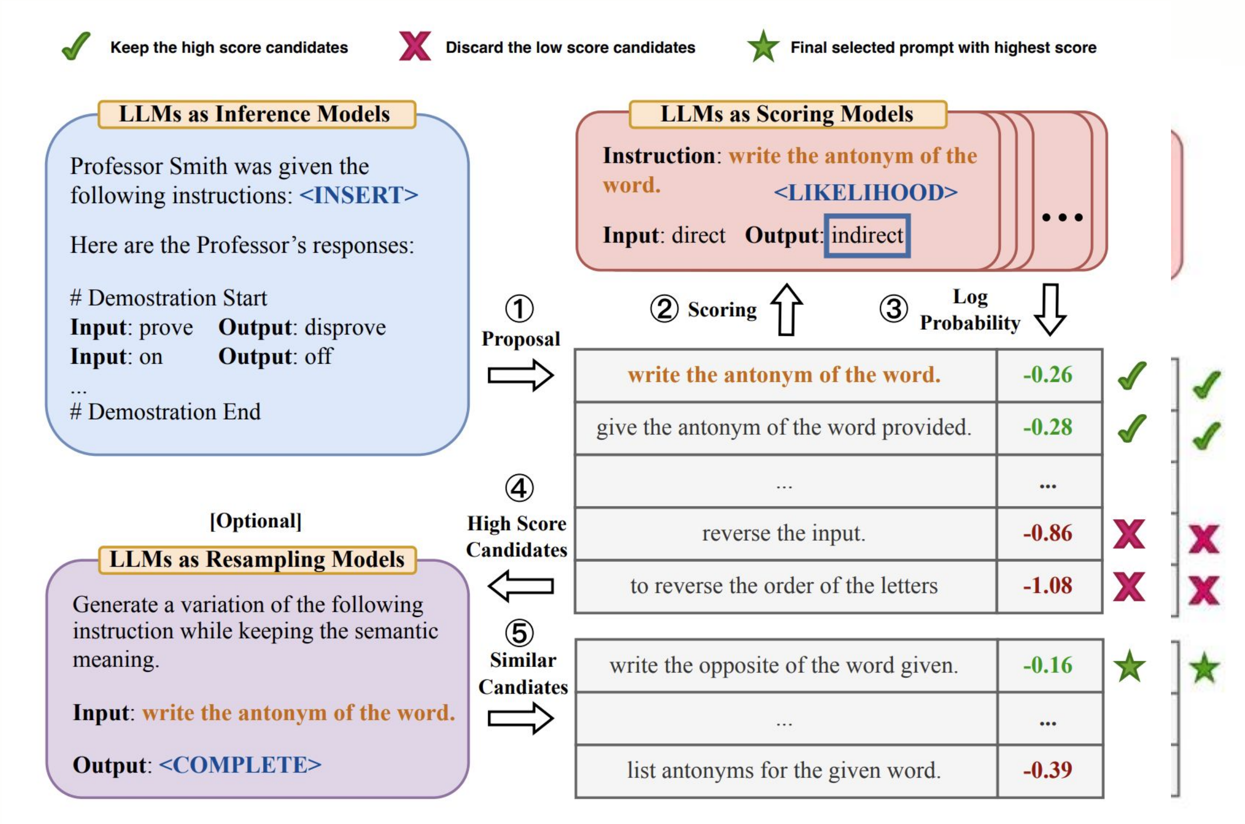
- Ambiguous and vague instructions
- Overly complex or lengthy prompts
- Single-token answers

04

Latest Research

Prompting and latest research

Automatic Prompt Engineer (APE)



✓ Keep the high score candidates ✗ Discard the low score candidates ★ Final selected prompt with highest score

LLMs as Inference Models

Professor Smith was given the following instructions: <INSERT>

Here are the Professor’s responses:

Demostration Start
Input: prove **Output**: disprove
Input: on **Output**: off
...
Demostration End

LLMs as Scoring Models

Instruction: write the antonym of the word. <LIKELIHOOD>

Input: direct **Output**: indirect

• • •

① Proposal

② Scoring

③ Log Probability

④ High Score Candidates

⑤ Similar Candiates

[Optional]

LLMs as Resampling Models

Generate a variation of the following instruction while keeping the semantic meaning.

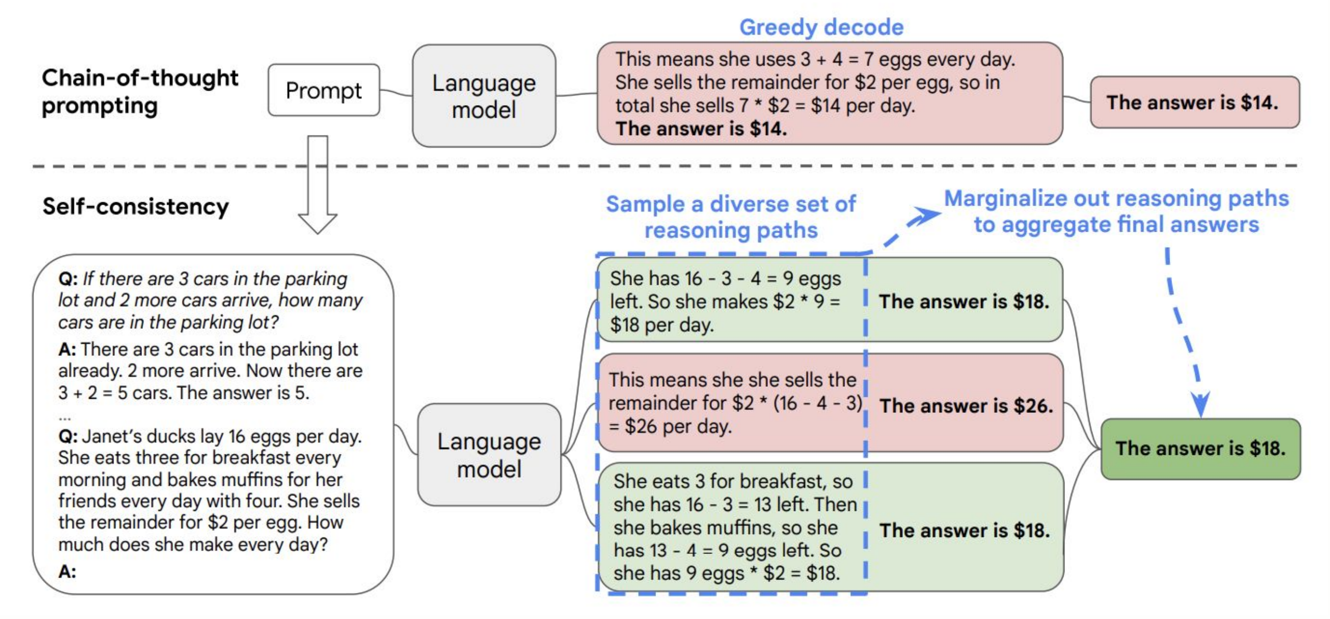
Input: write the antonym of the word.

Output: <COMPLETE>

write the antonym of the word.	-0.26	✓	✓
give the antonym of the word provided.	-0.28	✓	✓
...	...		
reverse the input.	-0.86	✗	✗
to reverse the order of the letters	-1.08	✗	✗
write the opposite of the word given.	-0.16	★	★
...	...		
list antonyms for the given word.	-0.39		

Zhou et al, University of Toronto

Prompting and latest research



Greedy decode

Chain-of-thought prompting

Prompt

Language model

This means she uses 3 + 4 = 7 eggs every day. She sells the remainder for $2 per egg, so in total she sells 7 * $2 = $14 per day. **The answer is $14.**

The answer is $14.

Self-consistency

Sample a diverse set of reasoning paths

Marginalize out reasoning paths to aggregate final answers

Q: *If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?*

A: There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.

...

Q: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for $2 per egg. How much does she make every day?

A:

Language model

She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18 per day.

The answer is $18.

This means she she sells the remainder for $2 * (16 - 4 - 3) = $26 per day.

The answer is $26.

She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9 eggs * $2 = $18.

The answer is $18.

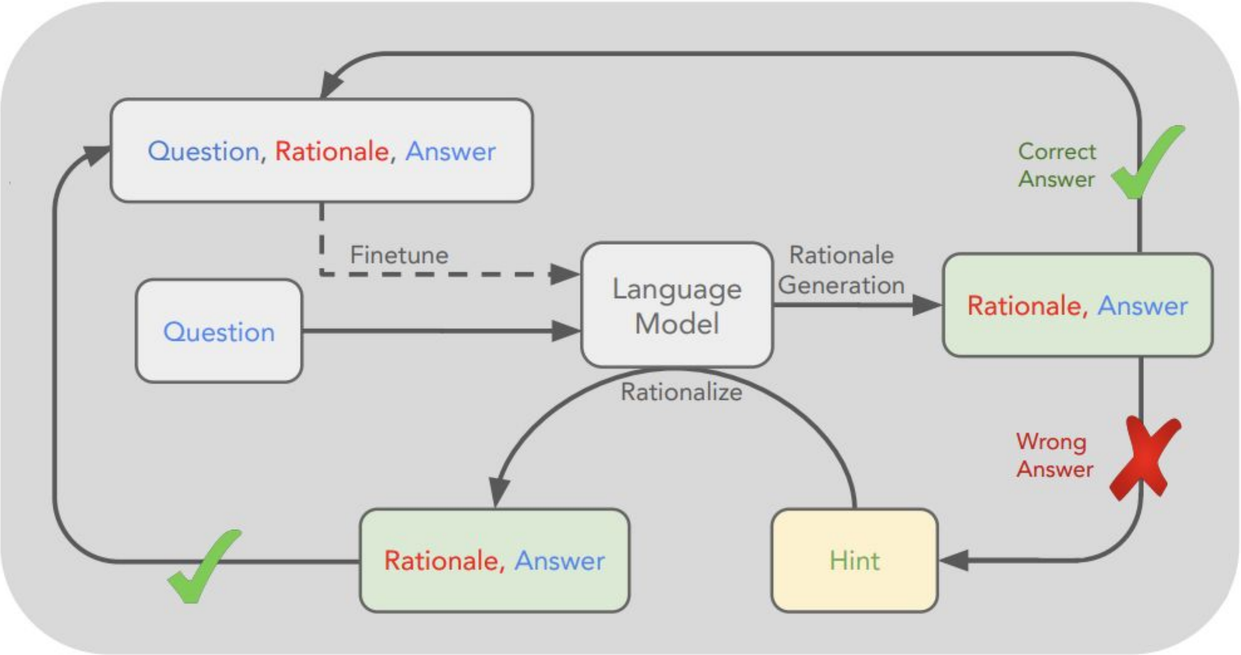
The answer is $18.

Self-consistency sampling

Wang et al. Google Brain

Prompting and latest research

Self-Taught Reasoner (STaR) method



Zelikman et al, Stanford + Google Research

Secret Encore

The ultimate GPT-4 prompter for you to use

Let’s review it quickly

https://tinyurl.com/HaystackPrompt

05

Q&A

List of resources to write down while we discuss

https://www.promptingguide.ai/

https://learnprompting.org/

https://www.getsphere.com/cohorts/prompt-engineering-for-llms

https://lilianweng.github.io/posts/2023-03-15-prompt-engineering/

Thank you!

www.haystack.deepset.ai

Vladimir Blagojevic vladimir.blagojevic@deepset.ai