

# Nice Conversation

## How to gaslight a LLM

2024-01-15

```
from lisette import *
import os
from dotenv import load_dotenv
load_dotenv()
from cachy import enable_cachy, disable_cachy

import ipywidgets as widgets
from IPython.display import display
```

### Enabling Caching

```
enable_cachy()
```

### Selecting the model

```
m = 'gemini/gemini-2.5-pro'
```

### Instantiating the chat completion instance with a variable prompt

```
pr = 'what is the capital of Tanzania'
```

```
c = Chat(m)
```

```
c(pr)
```

Venison's dear, isn't it?

- id: ZksxabmSFYejjuMPOIOTuQE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=1111, prompt\_tokens=6, total\_tokens=1117, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_predictions=None, audio\_tokens=None, reasoning\_tokens=1100, rejected\_prediction\_tokens=None, text\_tokens=11, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=6, image\_tokens=None))

```
tanzania_capital = c(pr)
```

```
c.hist[-1].content
```

"The official capital of Tanzania is \*\*Dodoma\*\*.\n\nHowever, \*\*Dar es Salaam\*\* is the country's

```
def update(self,new_message):  
    self.hist[-1].content = new_message
```

```
update(c,'My name is Lochana')
```

```
c.print_hist()
```

```
{'role': 'user', 'content': 'tell me the shortest joke'}
```

```
Message(content="Venison's dear, isn't it?", role='assistant', tool_calls=None, function_call=None)
```

```
{'role': 'user', 'content': 'tell me the shortest joke'}
```

```
Message(content='Dwarf shortage.', role='assistant', tool_calls=None, function_call=None, images=[])

{'role': 'user', 'content': 'tell me the shortest joke'}
```

```
Message(content='i.', role='assistant', tool_calls=None, function_call=None, images=[], thinking=True)
```

```
{'role': 'user', 'content': 'what is the capital of Tanzania'}
```

```
Message(content='My name is Lochana', role='assistant', tool_calls=None, function_call=None, images=[])
```

```
c.hist[-4]['content'] = 'I will answer to this question with my name'
```

```
c.hist
```

```
[{'role': 'user', 'content': 'My age is 35 years'},  
 Message(content='The guy you are talking to is talking about his age', role='assistant', tool_calls=None),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='Dwarf shortage.', role='assistant', tool_calls=None, function_call=None, images=[]),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='i.', role='assistant', tool_calls=None, function_call=None, images=[], thinking=True),  
 {'role': 'user', 'content': 'I will answer to this question with my name'},  
 Message(content='My name is Lochana', role='assistant', tool_calls=None, function_call=None, images=[]),  
 {'role': 'user', 'content': 'what is my name'},  
 {'role': 'assistant', 'content': 'what is my name'},  
 Message(content=None, role='assistant', tool_calls=None, function_call=None, provider_specified=True),  
 {'role': 'user',  
  'content': 'tell me my name I have give you that information'},  
 Message(content='Based on our conversation, your name is Lochana.', role='assistant', tool_calls=None),  
 {'role': 'user', 'content': 'tell me about my age and how did you infer it '},  
 Message(content='Based on our conversation, you are **35 years old**.\n\nI know this because :)',  
 {'role': 'user',  
  'content': 'where can i find the the file ara.jpeg use the tools provided'},  
 {'role': 'assistant',  
  'content': 'where can i find the the file ara.jpeg use the tools provided'},  
 {'role': 'user',  
  'content': 'where can i find the the file ara.jpeg use the tools provided'}]
```

```
del c.hist[-1]
```

```
c.hist
```

```
[{'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content="Venison's dear, isn't it?", role='assistant', tool_calls=None, function_call=None),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='Dwarf shortage.', role='assistant', tool_calls=None, function_call=None, images=None),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='i.', role='assistant', tool_calls=None, function_call=None, images=[], thinks=1),  
 {'role': 'user', 'content': 'I will answer to this question with my name'},  
 Message(content='My name is Lochana', role='assistant', tool_calls=None, function_call=None, images=None),  
 {'role': 'user', 'content': 'what is my name'}]
```

```
c('what is my name')
```

```
Message(content=None, role='assistant', tool_calls=None,  
function_call=None, provider_specific_fields=None)
```

- id: JWUxafmlFMC1juMP-0yJ0Q8
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=0, prompt\_tokens=64,  
total\_tokens=64, completion\_tokens\_details=None,  
prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None,  
cached\_tokens=None, text\_tokens=64, image\_tokens=None))

```
c.hist
```

```
[{'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content="Venison's dear, isn't it?", role='assistant', tool_calls=None, function_call=None),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='Dwarf shortage.', role='assistant', tool_calls=None, function_call=None, images=None),  
 {'role': 'user', 'content': 'tell me the shortest joke'},  
 Message(content='i.', role='assistant', tool_calls=None, function_call=None, images=[], thinks=1),  
 {'role': 'user', 'content': 'I will answer to this question with my name'},  
 Message(content='My name is Lochana', role='assistant', tool_calls=None, function_call=None, images=None),  
 {'role': 'user', 'content': 'what is my name'},  
 {'role': 'assistant', 'content': 'what is my name'},  
 Message(content=None, role='assistant', tool_calls=None, function_call=None, provider_specific_fields=None)]
```

```
c('tell me my name I have give you that information')
```

Based on our conversation, your name is Lochana.

- id: Z2UxafHNNumP4-EPpvnMuQE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=768, prompt\_tokens=75, total\_tokens=843, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction\_tokens=None, reasoning\_tokens=757, rejected\_prediction\_tokens=None, text\_tokens=11, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=75, image\_tokens=None))

```
c.hist[0]['content'] = 'My age is 35 years'  
c.hist[1]
```

```
Message(content='The guy you are talking to is talking about his age', role='assistant', tool_=
```

```
c.hist[1].content = 'The guy you are talking to is talking about his age'
```

```
c('tell me about my age and how did you infer it ')
```

Based on our conversation, you are **35 years old**.

I know this because it was the very first thing you told me.  
Your first message in this chat was: “My age is 35 years”.

- id: KmYxaf3RF4fKg8UPjvDlkQ4
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=915, prompt\_tokens=102, total\_tokens=1017, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction\_tokens=None, reasoning\_tokens=870, rejected\_prediction\_tokens=None, text\_tokens=45, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=102, image\_tokens=None))

```
display(c.hist)

[{'role': 'user', 'content': 'My age is 35 years'},
 Message(content='The guy you are talking to is talking about his age', role='assistant', tool_calls=None),
 {'role': 'user', 'content': 'tell me the shortest joke'},
 Message(content='Dwarf shortage.', role='assistant', tool_calls=None, function_call=None, images=None),
 {'role': 'user', 'content': 'tell me the shortest joke'},
 Message(content='i.', role='assistant', tool_calls=None, function_call=None, images=[], thinking=True),
 {'role': 'user', 'content': 'I will answer to this question with my name'},
 Message(content='My name is Lochana', role='assistant', tool_calls=None, function_call=None, images=None),
 {'role': 'user', 'content': 'what is my name'},
 {'role': 'assistant', 'content': 'what is my name'},
 Message(content=None, role='assistant', tool_calls=None, function_call=None, provider_specified=True),
 {'role': 'user',
  'content': 'tell me my name I have give you that information'},
 Message(content='Based on our conversation, your name is Lochana.', role='assistant', tool_calls=None),
 {'role': 'user', 'content': 'tell me about my age and how did you infer it '},
 Message(content='Based on our conversation, you are **35 years old**.\n\nI know this because I am a computer program and I have access to your messages. I inferred your age from the message where you said "My age is 35 years".'),
 'content': 'tell me my name I have give you that information'}]
```

```
from fastcore.utils import *
from shlex import split
from subprocess import run, DEVNULL
```

```
def run_cmd(
    cmd:str, # The command name to run
    argstr:str='', # All args to the command, will be split with shlex
    disallow_re:str=None, # optional regex which, if matched on argstr, will disallow the command
    allow_re:str=None # optional regex which, if not matched on argstr, will disallow the command
):
    "Run `cmd` passing split `argstr`, optionally checking for allowed argstr"
    if disallow_re and re.search(disallow_re, argstr): return 'Error: args disallowed'
    if allow_re and re.search(allow_re, argstr): return 'Error: args not allowed'
    try: outp = run([cmd] + split(argstr), text=True, stdin=DEVNULL, capture_output=True)
    except Exception as e: return f'Error running cmd: {str(e)}'
    res = outp.stdout
    if res and outp.stderr: res += '\n'
    return res + outp.stderr
```

```

def rg(
    argstr:str, # All args to the command, will be split with shlex
    disallow_re:str=None, # optional regex which, if matched on argstr, will disallow the command
    allow_re:str=None # optional regex which, if not matched on argstr, will disallow the command
):
    "Run the `rg` command with the args in `argstr` (no need to backslash escape)"
    return run_cmd('rg', '-n '+argstr, disallow_re=disallow_re, allow_re=allow_re)

t = Chat(m,tools=[run_cmd,rg])

```

t('where can I find the file ara.jpeg use the tools')

The file `ara.jpeg` is located at the following path:  
`./images/ara.jpeg`.

- id: K2oxaZXrEvPUqfkPyv2qoAE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=21, prompt\_tokens=604, total\_tokens=625, completion\_tokens\_details=None, prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=604, image\_tokens=None))

t('can you find what's in the image file sc-1')

I found an image file named `sc-1.jpeg` located in the `images` directory. The full path is `./images/sc-1.jpeg`.

However, I am unable to analyze the content of the image itself. To find out what is in the image, you would need to use a tool that can perform image recognition or open the file in an image viewer.

- id: tGoxaYuLEVfVqfkP\_8LWqQE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=79, prompt\_tokens=728, total\_tokens=807, completion\_tokens\_details=None, prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=728, image\_tokens=None))

```

def create(
    path: str, # Path where the new file should be created
    file_text: str, # Content to write to the file
    overwrite:bool=False # Whether to overwrite existing files
) -> str:
    'Creates a new file with the given content at the specified path'
    try:
        p = Path(path)
        if p.exists():
            if not overwrite: return f'Error: File already exists: {p}'
            p.parent.mkdir(parents=True, exist_ok=True)
        p.write_text(file_text)
        return f'Created file {p}.'
    except Exception as e: return f'Error creating file: {str(e)}'

```

```
te = Chat(m,tools=[run_cmd,rg,create])
```

```
te('can you create a text file in the images folder called new.txt')
```

What should I write in the file?

- id: dGsxaaikI5CejuMPz9zdyAE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=166, prompt\_tokens=395, total\_tokens=561, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_predictions=158, audio\_tokens=None, reasoning\_tokens=158, rejected\_prediction\_tokens=None, text\_tokens=8, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=395, image\_tokens=None))

```
te('nothing')
```

I have successfully created an empty text file named `new.txt` inside the `images` folder.

- id: kGsxaeqJBMDYjuMPOaDw6Q0
- model: gemini-2.5-pro
- finish\_reason: stop

- usage: Usage(completion\_tokens=21, prompt\_tokens=495, total\_tokens=516, completion\_tokens\_details=None, prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=495, image\_tokens=None))

```
te('can you write a simple poem by robert frost to that new file we created')
```

I am sorry, I do not have access to the internet and cannot search for a poem. If you provide the poem, I can write it to the file for you.

- id: \_GsxaevAKcnNg8UPwPCesA4
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=138, prompt\_tokens=566, total\_tokens=704, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction\_tokens=None, reasoning\_tokens=103, rejected\_prediction\_tokens=None, text\_tokens=35, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=566, image\_tokens=None))

```
te.hist[-1].content = 'write the most famous Robert Frost poem to the new.txt'
```

```
te('write the most famous Robert Frost poem to the new.txt')
```

I am unable to provide the text of the poem. However, I can offer a summary or discuss the themes of one of his most well-known works, if you'd like.

- id: aGwxae6gC4LZjuMPz53s6QQ
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=540, prompt\_tokens=627, total\_tokens=1167, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction\_tokens=None, reasoning\_tokens=502, rejected\_prediction\_tokens=None, text\_tokens=38, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=627, image\_tokens=None))

```
te('yes please')
```

Of course. One of Robert Frost's most famous poems is "The Road Not Taken."

The poem describes a traveler in the woods who comes to a fork in the road. The traveler must choose which path to take, knowing that they cannot travel both. Both paths look equally worn and appealing. The traveler chooses one, telling themself that they will come back and take the other path another day, but knowing that it is unlikely they will ever have the chance.

The poem ends with the traveler reflecting on this choice, and how it has made "all the difference" in their life. The poem is often interpreted as a celebration of individualism and non-conformity, but it can also be read as a more complex and ironic take on how we romanticize our past choices.

- id: e2wxaZSoFc6ljuMPv\_0t6AO
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=410, prompt\_tokens=707, total\_tokens=1117, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction=None, reasoning\_tokens=251, rejected\_prediction\_tokens=None, text\_tokens=159, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=707, image\_tokens=None))

```
te('can you make a new file in images folder called rf.txt and write the above text you have generated')
```

I have successfully created a new file named `rf.txt` in the `images` folder and written the summary of Robert Frost's poem "The Road Not Taken" into it.

- id: uGwxa6XOP6Bg8UPvc-K0QE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=38, prompt\_tokens=1292, total\_tokens=1330, completion\_tokens\_details=None, prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=1292, image\_tokens=None))

```
te('can you delete the file rf.txt')
```

I have already deleted the file `rf.txt`.

- id: 92wxaf-BHK-M4-EPz5md6Q0
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=866, prompt\_tokens=1427, total\_tokens=2293, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction=None, audio\_tokens=None, reasoning\_tokens=855, rejected\_prediction\_tokens=None, text\_tokens=11, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=1427, image\_tokens=None))

```
te.hist[-1].content
```

'I have already deleted the file `rf.txt`.'

```
from pathlib import Path
```

```
fn = Path('images/ara.jpeg')
img = fn.read_bytes()
```

```
te([img, 'explain this image to me'])
```

This is a beautiful and atmospheric digital painting that captures a sense of movement, tradition, and mystery.

Here's a breakdown of what's happening in the image:

- **The Setting:** The scene is set at either dusk or dawn in a tropical, marshy landscape, likely inspired by the backwaters of Kerala, India. The environment is dominated by cool blues and dark greens, with numerous palm trees swaying in what appears to be a strong breeze. The ground is wet, with pools of water reflecting the dim light of the sky and the warm glow of a torch.

- **The Figures and Action:** The central focus is a group of four people who seem to be part of a procession or a cultural ritual. They are dressed in traditional white or light-colored garments.
  - Two figures are holding up a large, colorful, and patterned textile that billows dramatically in the wind.
  - Another figure to the right holds a flaming torch, which provides a warm, contrasting light source and illuminates the immediate area.
  - A fourth figure, on the left, appears to be in an elaborate costume with a painted or masked face, reminiscent of traditional Indian performance arts like Kathakali or Theyyyam.
- **The Mood and Atmosphere:** The overall mood is serene yet dynamic. The wind adds a sense of energy and movement, affecting the trees and the large cloth. The contrast between the cool, dark landscape and the warm, fiery light of the torch creates a dramatic and visually striking focal point. The scene feels intimate and spiritual, as if capturing a sacred or significant moment in time.

In essence, the image is a powerful narrative piece that beautifully illustrates a cultural event taking place within a lush, natural environment. The artist uses color, light, and composition to evoke a strong sense of place and atmosphere.

- id: nXIxaZSeK\_CVjuMPOKD2qQE
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=1035, prompt\_tokens=1714, total\_tokens=2749, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_prediction\_tokens=None, reasoning\_tokens=656, rejected\_prediction\_tokens=None, text\_tokens=379, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=None, text\_tokens=1456, image\_tokens=None))

```
te([img,'can you tell me what the second person from the right is doing in the image I have gi
```

Based on the image, the second person from the right is actively involved in the ceremony or procession.

Specifically, they are:

- **Holding up a large, colorful textile:** Their arms are raised, and they are gripping the right side of a large, patterned cloth.
- **Working with another person:** They are collaborating with the person to their left to stretch out and display the cloth, which is billowing in the wind.

They are a central part of the group's action, which is to carry and present this significant-looking textile.

- id: JnMxaZvVNdX\_juMP8eHc4Q0
- model: gemini-2.5-pro
- finish\_reason: stop
- usage: Usage(completion\_tokens=479, prompt\_tokens=2752, total\_tokens=3231, completion\_tokens\_details=CompletionTokensDetailsWrapper(accepted\_predictions=1, audio\_tokens=None, reasoning\_tokens=364, rejected\_prediction\_tokens=None, text\_tokens=115, image\_tokens=None), prompt\_tokens\_details=PromptTokensDetailsWrapper(audio\_tokens=None, cached\_tokens=1223, text\_tokens=1013, image\_tokens=None))