



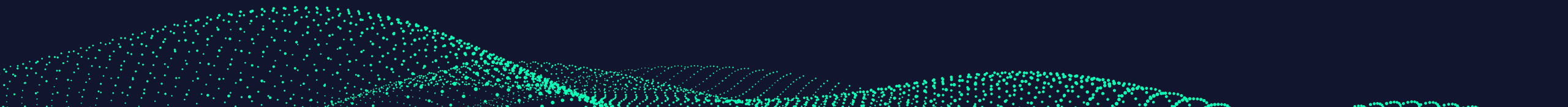
Is **Prompting** Enough?

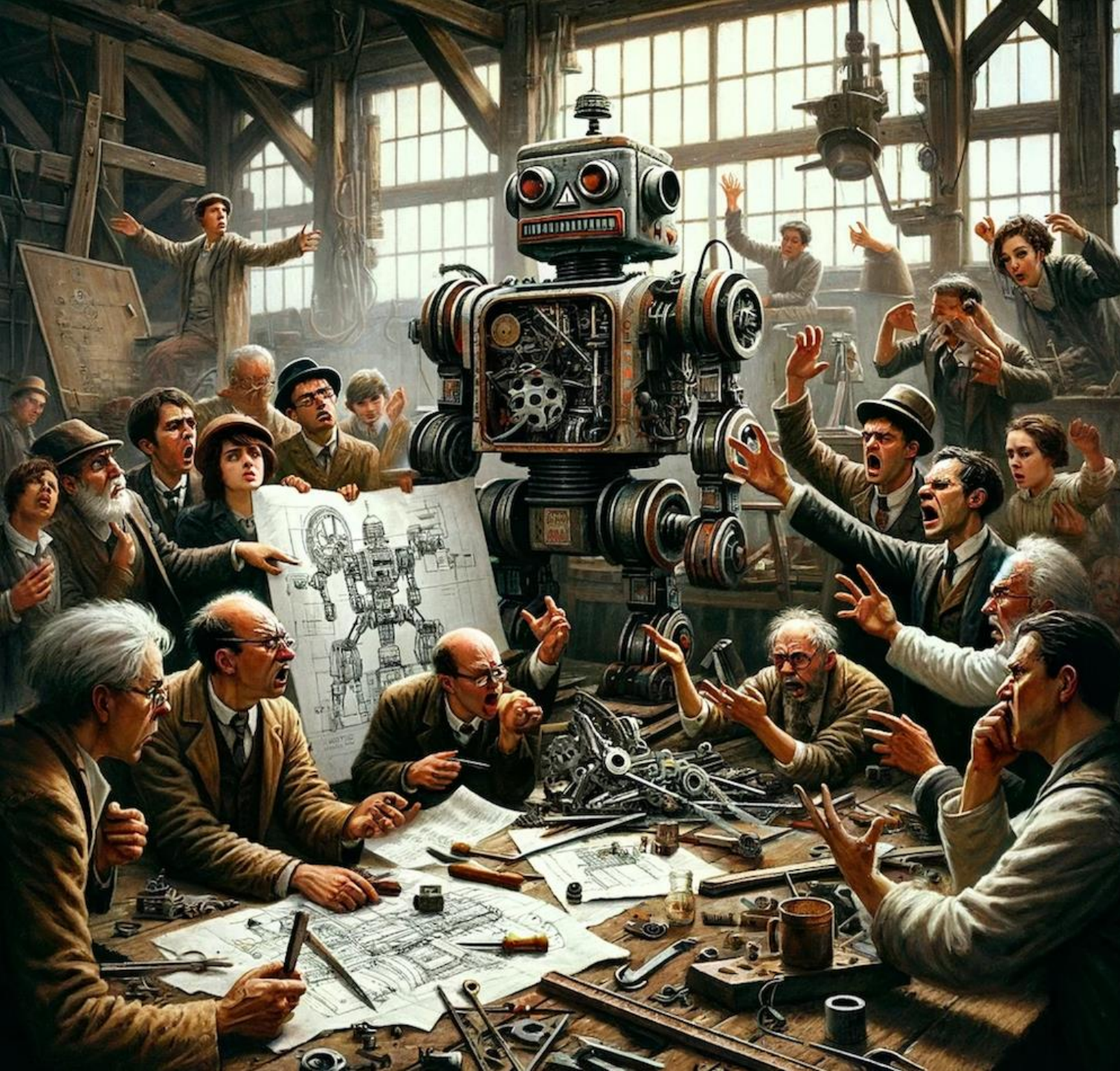
The Process of **Making a Copilot** for UI-based Chatbot Builders

Emanuel Lacić

Principal Engineer


Infobip





Trend 23/24:
Companies everywhere are
launching **Copilots**

AI assistants that
leverage **LLMs** to
perform actions



Copilot

Get answers to complex questions

For example, you could ask "Help me plan for my fishing trip."


Take actions on your PC

Control your Windows environment with actions like "Adjust my settings so I can focus."


Work across documents

Summarize and compose text from any app - start by copying text to clipboard.

Let's learn together. Windows copilot is powered by AI, so surprises and mistakes are possible. Make sure to check the facts, and share feedback so we can learn and improve!






Ask me anything...








11:11 AM
5/23/2023









Document1 · Saved

File Home Insert Layout References Review View Help





Aptos (Body) 11






Create content with Copilot

draft a proposal from yesterday's meeting notes






MS Office Copilot

Windows Copilot

Midjourney


<https://s.mj.run/wizde3mU47l> <https://s.mj.run/STFZLGUVVxA> An illustration, unique and colourful, A long shot of a dreamy land, a girls is floating in the air, She is happily looking at the photoframes floating around her, bold and pleasant colours, 8k, cinematic, detailed, unreal engine, --ar 2:1 --v 5 - @stashlers (relaxed)




U1 U2 U3 U4

V1 V2 V3 V4

GITHUB COPILOT: CHAT

 Write a unit test for this function



```
import unittest
import datetime

class TestParseExpenses(unittest.TestCase):
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n2023-01-03 2.59 DKK\n2023-01-03 -2.72 EUR'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'), (datetime.datetime(2023, 1, 3), 2.59, 'DKK'), (datetime.datetime(2023, 1, 3), -2.72, 'EUR')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)

if __name__ == '__main__':
    unittest.main()
```

The code assumes that the datetime module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.

Ask a question or type '/' for commands

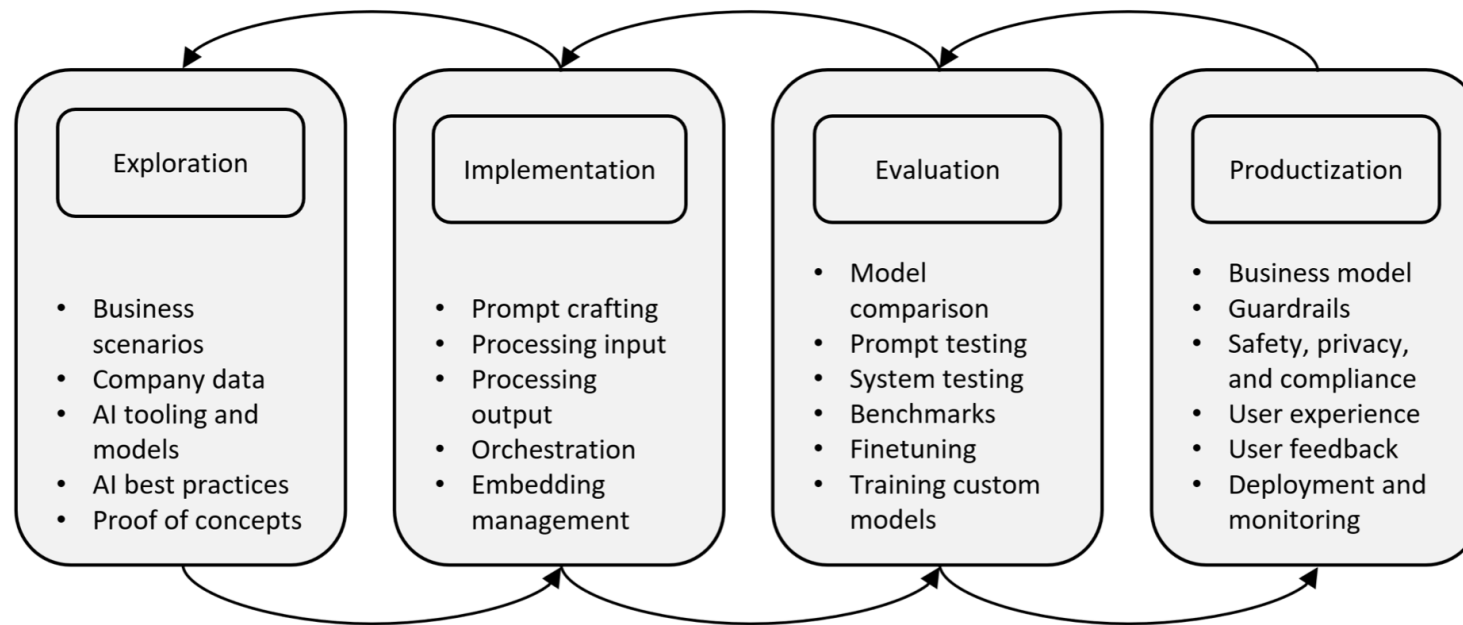
module.py

```
1 import datetime
2
3 def parse_expenses (expenses_string):
4     """Parse the list of expenses and return the list of triples (date, amount, currency)
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2023-01-02 -34.01 USD
9         2023-01-03 2.59 DKK
10        2023-01-03 -2.72 EUR
11
12    """
13    expenses = []
14
15    for line in expenses_string.splitlines():
16        if line.startswith("#"):
17            continue
18        date, value, currency = line.split(" ")
19        expenses.append((datetime.datetime.strptime (date, "%Y-%m-%d"),
20                        float (value),
21                        currency))
22
23    return expenses
24
25 expenses_data = '''2023-01-02 -34.01 USD
26                  2023-01-03 2.59 DKK
27                  2023-01-03 -2.72 EUR'''
```

Github Copilot

Building Your Own Product Copilot: Challenges, Opportunities, and Needs

Chris Parnin, Gustavo Soares, Rahul Pandita, Sumit Gulwani, Jessica Rich, Austin Z. Henley
{chrisparnin,gustavo.soares}@microsoft.com,rahulpandita@github.com,{sumitg,jessrich,austinhenley}@microsoft.com
Microsoft, GitHub Inc.
USA



Dialogs ⓘ



Default



Default

Welcome

Menu

+ ADD DIALOG

+ ADD GROUP

Info



Share some info

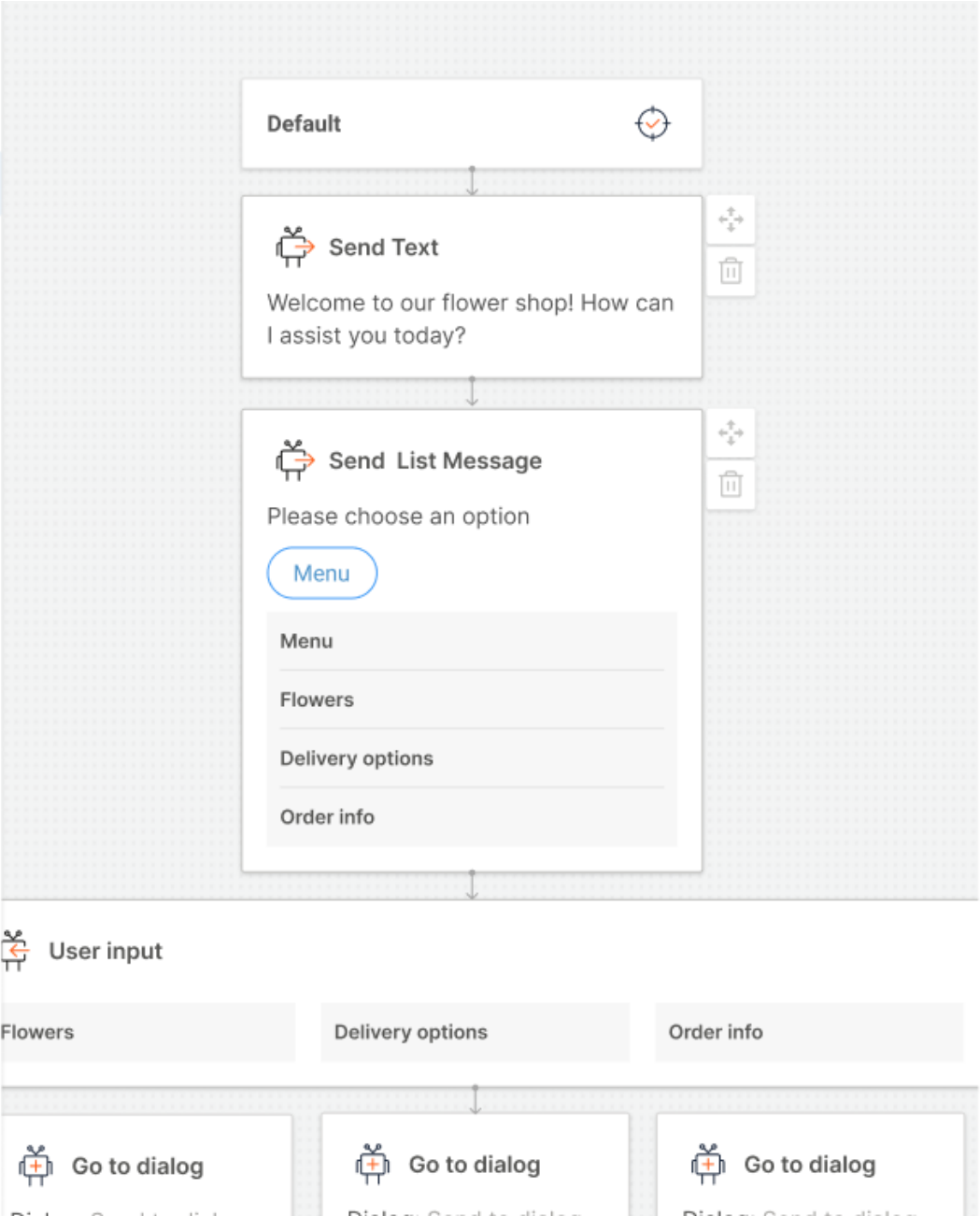
Get some info

Process info

Finish

+ ADD DIALOG

+ ADD GROUP



Build

Drag and drop the following elements to build and define your bot interactions or choose to build with AI copilot.

BUILD WITH AI COPILOT

Chatbot sends

Text

Image

Audio

File

Video

Location

Reply button

List

Sticker

Chatbot receives

Create dialog using Answers CoPilot



Explain what you want this dialog to do or what it needs to contain and Answers CoPilot will create it for you. [Learn more](#)

Name your dialog (optional)

Feedback dialog

Describe the dialog

0/160

Select mode of response ⓘ

Standard



Expect to see common chat design patterns that are proven to work. You can rely on this mode to always generate consistent results.

Model ⓘ

GPT-3



CANCEL

CREATE DIALOG

Create dialog using Answers CoPilot



Explain what you want this dialog to do or what it needs to contain and Answers CoPilot will create it for you. [Learn more](#)

Name your dialog (optional)

Feedback dialog

Describe the dialog

0/160

Select mode of response ⓘ

Standard



Expect to see common chat design patterns that are proven to work. You can rely on this mode to always generate consistent results.

Model ⓘ

GPT-3



CANCEL

CREATE DIALOG

Image Response



Code

```
var lastMsg = attributeApi.get('lastf
```

...



Call API

POST <https://image2text.ib-inet.com/img/caption>



Send Text

Looking for visually similar products from catalog...



Send Text

Create dialog using Answers CoPilot



Explain what you want this dialog to do or what it needs to contain and Answers CoPilot will create it for you. [Learn more](#)

Name your dialog (optional)

Feedback dialog

Describe the dialog

0/160

Select mode of response ⓘ

Standard



Expect to see common chat design patterns that are proven to work. You can rely on this mode to always generate consistent results.

Model ⓘ

GPT-3



CANCEL

CREATE DIALOG

Prompting with
commercial LLMs

Visual
Elements
[...]



vs



Image Response



Code

```
var lastMsg = attributeApi.get('lastf
```

...



Call API

```
POST https://image2text.ib-inet.com/imgf  
/caption
```



Send Text

Looking for visually similar products from
catalog... 🤖



Send Text

Prompting

Test out **prompt techniques** with commercial LLMs

A prompt that describes the problem of building a chatbot dialog as well as states the vocabulary of the available visual elements

Zero-Shot

Add the information about specific rules that need to be enforced to render the generated output in the UI

Few-Shot with Instructions

Few-Shot

Add multiple examples of input task descriptions and their expected outputs

Chain-of-Thought

Add reasoning steps, including how to understand the input, identify main actions, select and sequence visual elements, etc.

Performance

Hallucinations: Percentage of **predictions that contain hallucinations**. Hallucinations are unexpected predictions which include (1) format validation, (2) vocabulary validation and (3) rule validation

HitRate: Is 1 when the prediction **100% matches what is expected**, else 0

	Hallucinations	HitRate	
Zero-Shot	92.60 %	5.00 %	Google's Gemini 1.0 Pro
Few-Shot	64.73 %	3.84 %	
Few-Shot with Instructions	91.77 %	1.84 %	
CoT	9.06 %	5.41 %	
Zero-Shot	46.44 %	2.09 %	OpenAI's GPT-3.5
Few-Shot	12.63 %	1.75 %	
Few-Shot with Instructions	25.70 %	0.69 %	
CoT	9.17 %	3.68 %	



Adapting LLMs

OpenAI GPT3.5-turbo (**large**)

<https://learn.microsoft.com/en-us/azure/ai-services/openai/tutorials/fine-tune>

Mistral 7B Instruct (**mid**)

<https://arxiv.org/pdf/2310.06825.pdf>

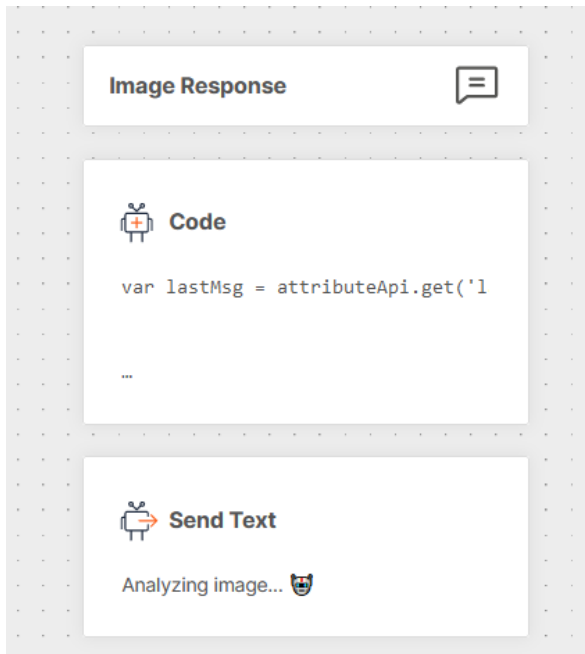
LLaMa 3B (**small**)

<https://arxiv.org/pdf/2302.13971.pdf>

Sheared LLaMA 1.3B (**tiny**)

<https://arxiv.org/pdf/2310.06694.pdf>

Training Data



Input	Output
A location-based financial service that allows users to transfer money, check wallet status, and find nearby branches.	[...]
Multi-functional tool that assists users with money transfers, currency exchange, and locating branches of a specific business.	[...]
Create a location-based service that helps users find branches, transfer money, and exchange currency.	[...]

What does this
dialog do?

Domain-specific
configuration of visual elements

Training Data



Input	Output
A location-based financial service that allows users to transfer money, check wallet status, and find nearby branches.	[...]
Multi-functional tool that assists users with money transfers, currency exchange, and locating branches of a specific business.	[...]
Create a location-based service that helps users find branches, transfer money, and exchange currency.	[...]

**BUT WE DON'T HAVE
THIS KIND OF DATA !!!**

Synthetic Data

Hypothesis: You can use GenAI to synthetically create **description data**

```
import json

instruction = """
You are a chabtot generator. Your job is to find out and describe what a bot is based on the provided attributes.
"""

prompt = """
You just got the following information about the attributes of the chatbot which will be built:

{attributes}

Describe in one sentence what this chatbot is about?
"""


def parse_json(json_str):
    attributes = []
    try:
        for obj in json.loads(json_str):
            attributes.append(obj["name"])
    except json.JSONDecodeError:
        return None

    messages=[
        {"role": "system", "content": instruction},
        {"role": "user", "content": prompt.format(attributes=attributes)}
    ]

    bot_desc = chat_complete(messages, temperature=0.0)

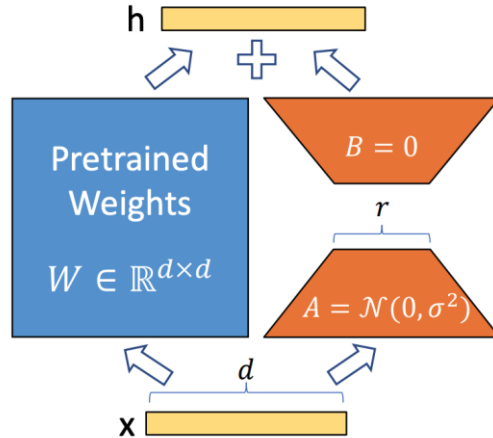
    return bot_desc
```

**Need for prior
data cleaning,
text standardization,
anonymization
& prompt engineering**



Fine-Tuned Models

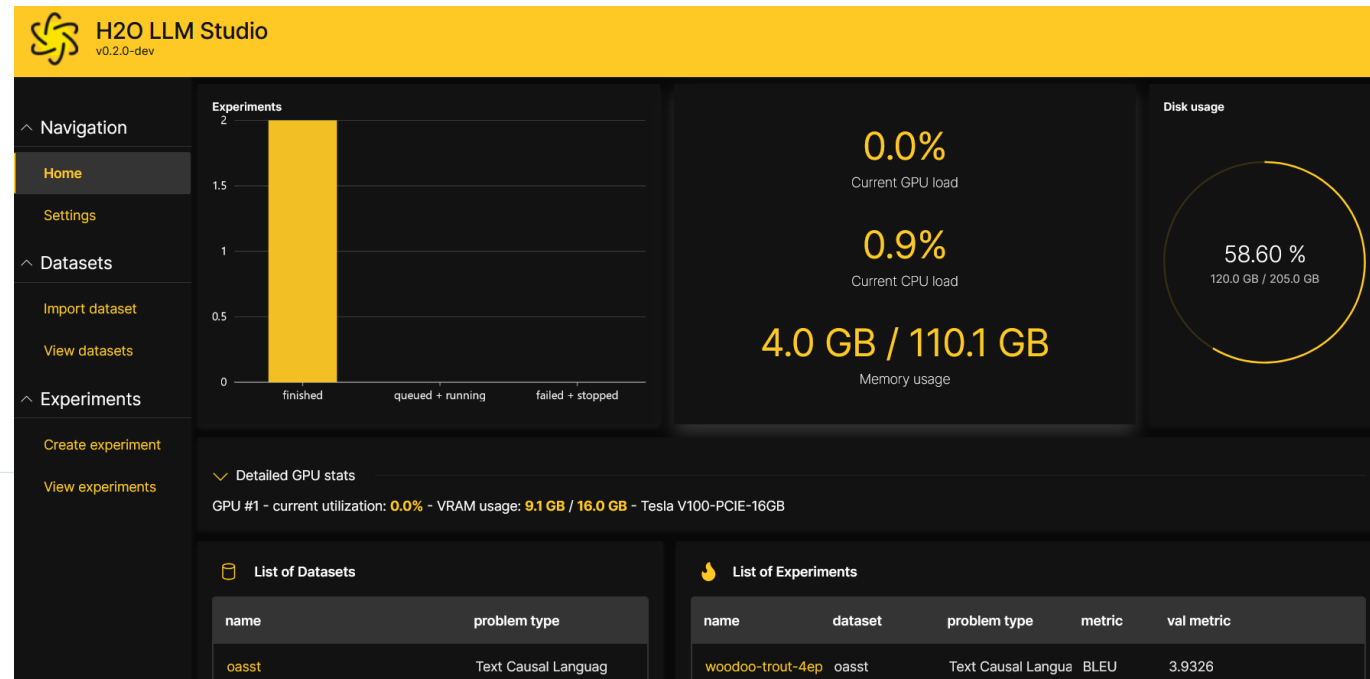
Use **LoRA** to fine-tune visual element generation on own data



State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods

<https://github.com/huggingface/peft>

<https://github.com/h2oai/h2o-llmstudio>



Create dialog using Answers CoPilot



Explain what you want this dialog to do or what it needs to contain and Answers CoPilot will create it for you. [Learn more](#)

Name your dialog (optional)

Feedback dialog

Describe the dialog

0/160

Collect feedback from users about their experience with a product or service, prompting them to rate and provide comments.

Select mode of response ⓘ

Standard



Expect to see common chat design patterns that are proven to work. You can rely on this mode to always generate consistent results.

Model ⓘ

GPT-3



CANCEL

CREATE DIALOG

Fine-tuned LLMs
on relevant data

Visual Elements
[...]

Image Response



Code

```
var lastMsg = attributeApi.get('lastf
```

...



Call API

POST https://image2text.ib-inet.com/img/caption



Send Text

Looking for visually similar products from catalog...



Send Text

Fine-Tuned Models

Number of Hallucinations significantly lowered from 9.06% → the **best performance of 0.04%**
A HitRate that improved from 0.69% - 5.41% → **18.81% - 26.72%**

	Hallucinations	HitRate
Sheared LLaMA 1.3B (tiny)	0.04 %	18.81 %
LLaMa 3B (small)	0.19 %	18.89 %
Mistral 7B Instruct (mid)	15.34 %	26.72 %
OpenAI GPT3.5-turbo (large)	1.96 %	15.78 %

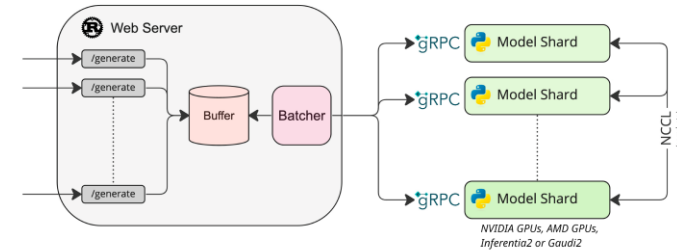
Inference

For inference, you can use Huggingface's text generation API

<https://github.com/huggingface/text-generation-inference>

Text Generation Inference

Fast optimized inference for LLMs



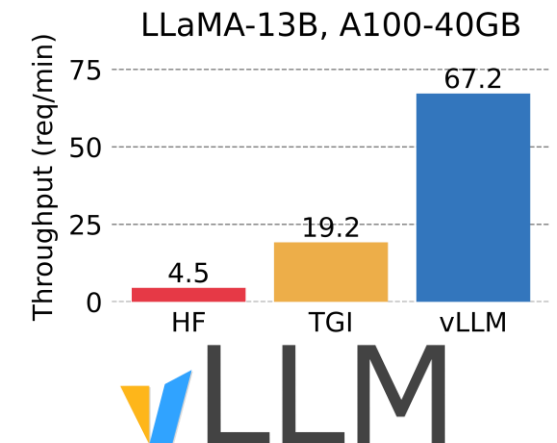
```
docker run --detach --gpus all --shm-size 1g -p 9999:80 -v /var/lib/docker/volumes/h2o-llmstudio-shared/output/user:/data
ghcr.io/huggingface/text-generation-inference:1.1.0 --model-id /data/mymodel
```

Mistral-7B on NVIDIA's Volta architecture requires the use of **llama.cpp**

<https://github.com/ggerganov/llama.cpp>

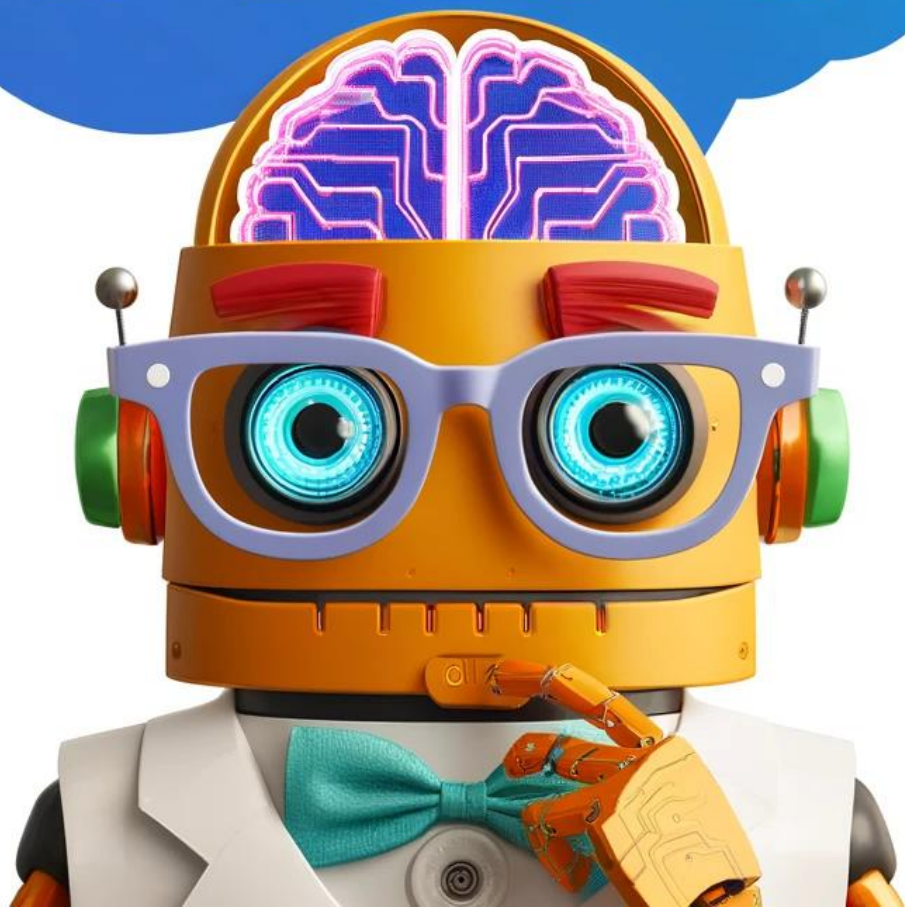
LLaMA C++

	VRAM
Sheared LLaMA 1.3B (tiny)	5.1 GB
LLaMa 3B (small)	9.5 GB
Mistral 7B Instruct (mid)	13.6 GB



<https://github.com/vllm-project/vllm>

NEXT STEPS



Create dialog using Answers CoPilot

Explain what you want this dialog to do or what it needs to contain and Answers CoPilot will create it for you. [Learn more](#)

Name your dialog (optional)

Feedback dialog

Describe the dialog

0/160

Collect feedback from users about their experience with a product or service, prompting them to rate and provide comments.

Select mode of response ⓘ

Standard

Expect to see common chat design patterns that are proven to work. You can rely on this mode to always generate consistent results.

Model ⓘ

GPT-3

CANCEL

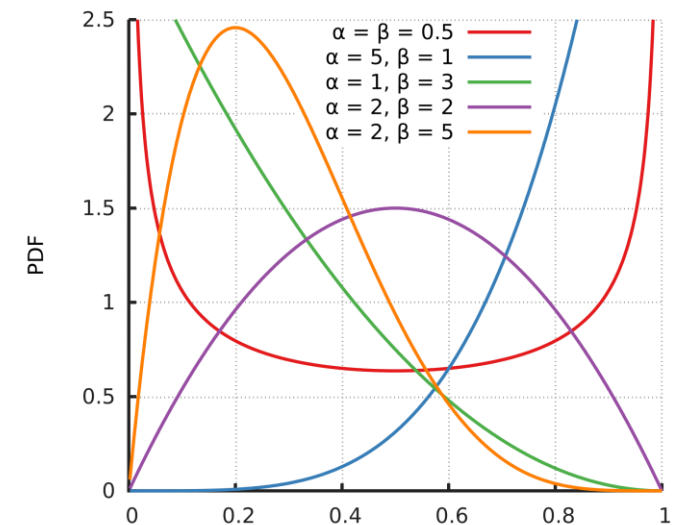
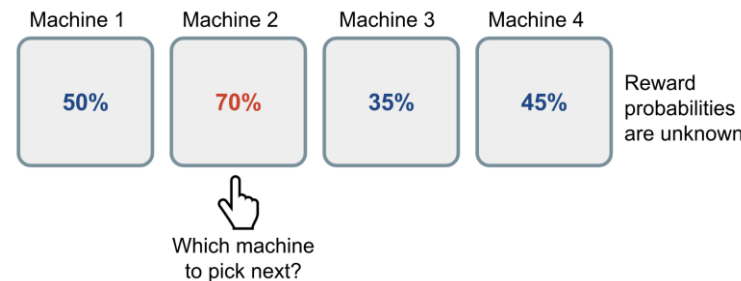
CREATE DIALOG

POSITION BIAS

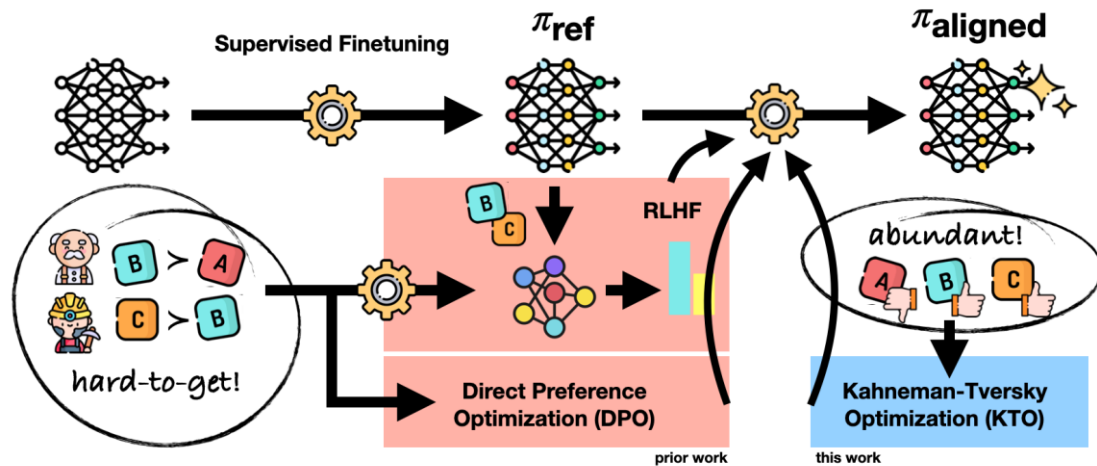
90% of the users stuck with the **default model**

87% used the **default temperature**, preferring to rephrase queries over changing settings when errors occurred

To reduce user burden, should implement an online model **selection algorithm** (e.g., time-increasing **bandit algorithm**) for better usability



Human-Aware Loss Functions



<https://github.com/ContextualAI/HALOs>

KTO: Model Alignment as Prospect Theoretic Optimization

Kawin Ethayarajh¹ Winnie Xu² Niklas Muennighoff² Dan Jurafsky¹ Douwe Kiela^{1,2}

Abstract

Kahneman & Tversky's *prospect theory* tells us that humans perceive random variables in a biased but well-defined manner (1992); for example, humans are famously loss-averse. We show that objectives for aligning LLMs with human feedback implicitly incorporate many of these biases—the success of these objectives (e.g., DPO) over cross-entropy minimization can partly be ascribed to them being *human-aware loss functions* (HALOs). However, the utility functions these meth-

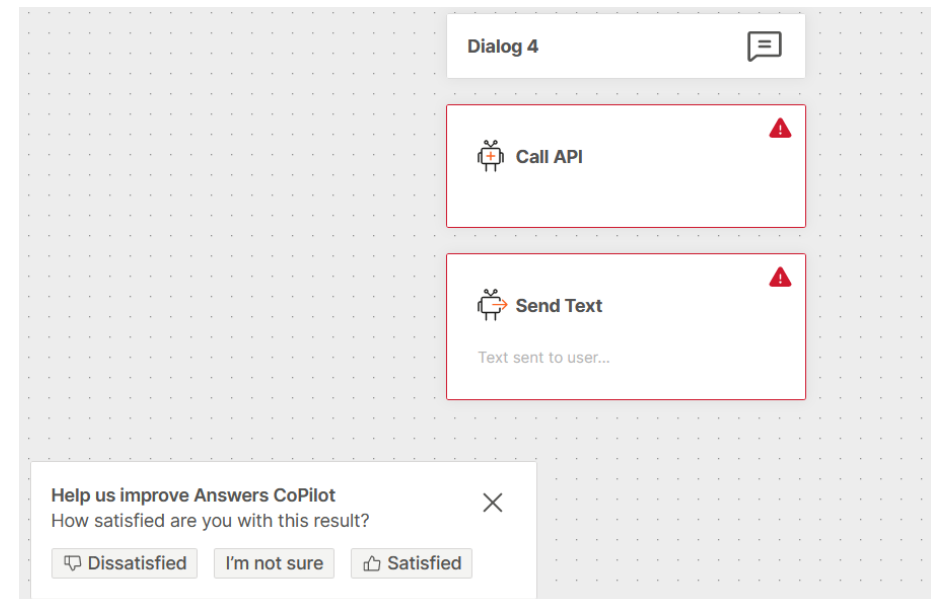
the mathematically equivalent DPO (Rafailov et al., 2023)—take preference data as input.

To understand why these alignment methods work so well, and whether feedback needs to be in the form of preferences, we frame them through the lens of *prospect theory* (Kahneman & Tversky, 1979; Tversky & Kahneman, 1992). Prospect theory explains why humans make decisions about uncertain events that do not maximize expected value. It formalizes how humans perceive random variables in a biased but well-defined manner; for example, relative to some reference point, humans are more sensitive to losses

Human feedback is in a **binary format**?

There is an **imbalance** between the number of **desirable** and **undesirable examples**?

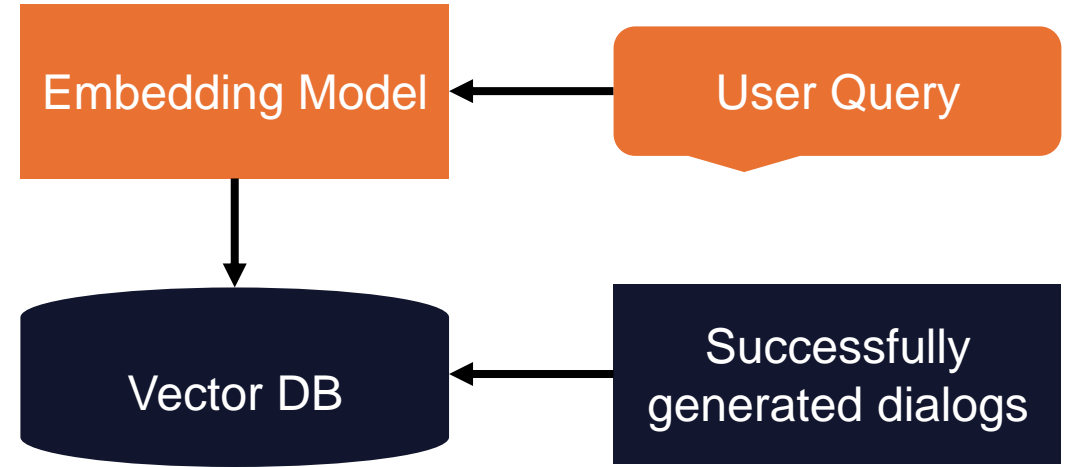
In that case, KTO is the natural choice!



User Frustration

Hallucination has to be (almost) 0%

Always have a **simpler fallback option**
like a **kNN approach for retrieving** successfully
generated dialogs that are similar to the user query

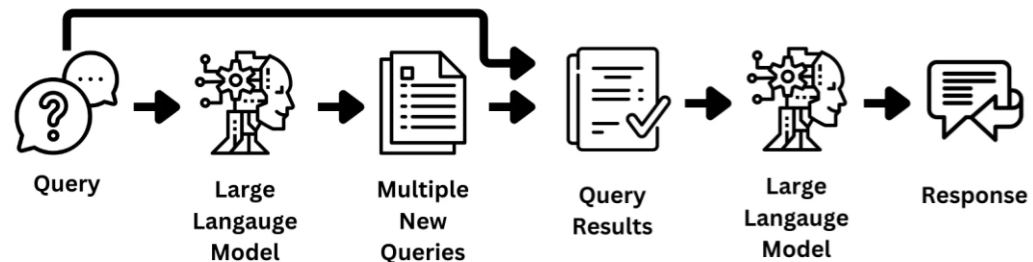


Explainability and Control

Lack of **explanations** for **failures** or logging the **historical output** hinders user acceptance and trust
UX/UI design plays an important role as well as splitting tasks (e.g., llama agents: generation vs. FAQ)

Handling short queries

Adapt the input with **Query Expansion**



Thank you!



Emanuel Lacić
Principal Engineer @ **Infobip**

 emanuel.lacic@infobip.com

 [@elacic1](https://twitter.com/elacic1)

 [/in/elacic](https://www.linkedin.com/in/elacic)

 <http://elacic.me>