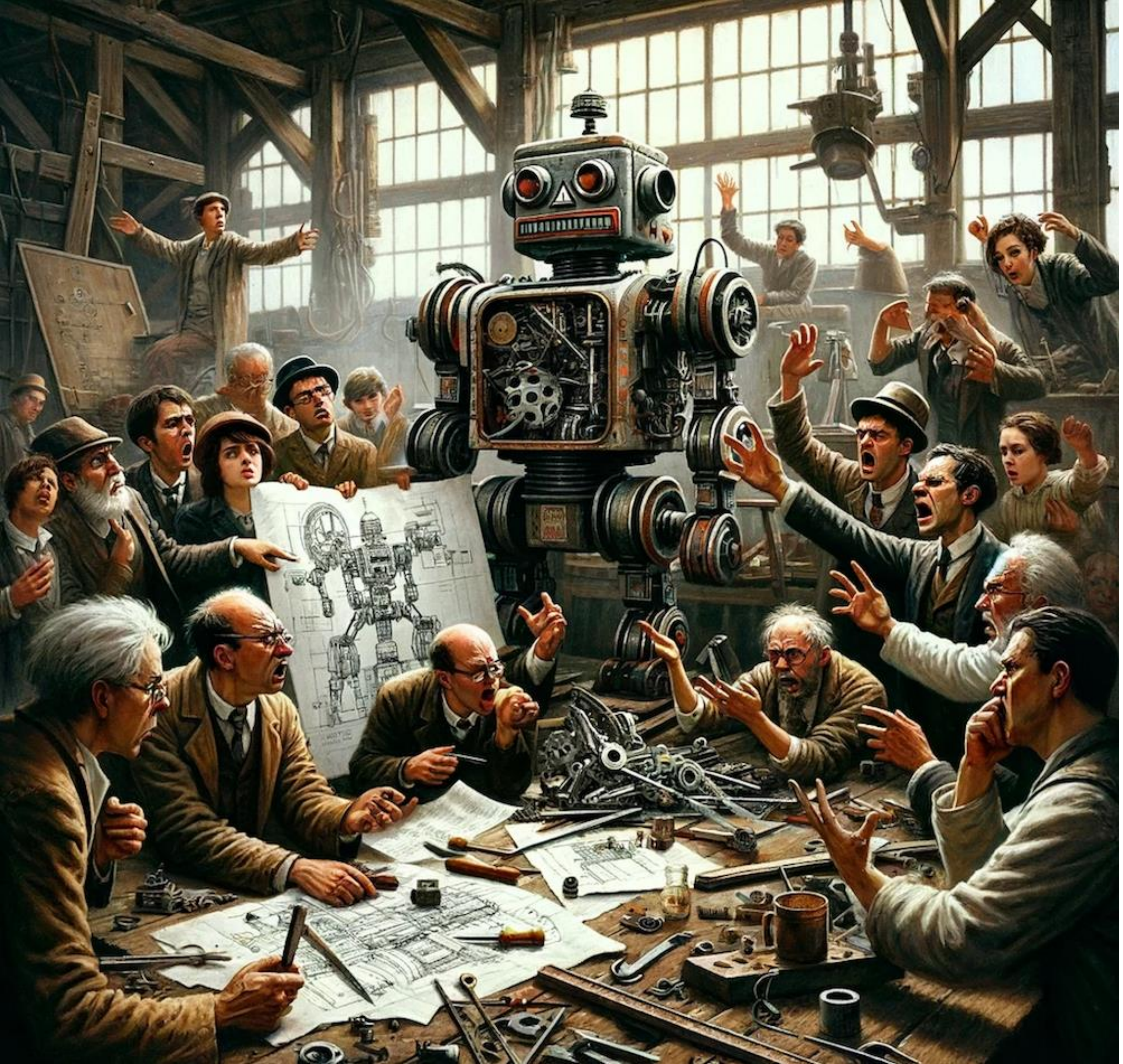
Is Prompting Enough?

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

Emanuel Lacić

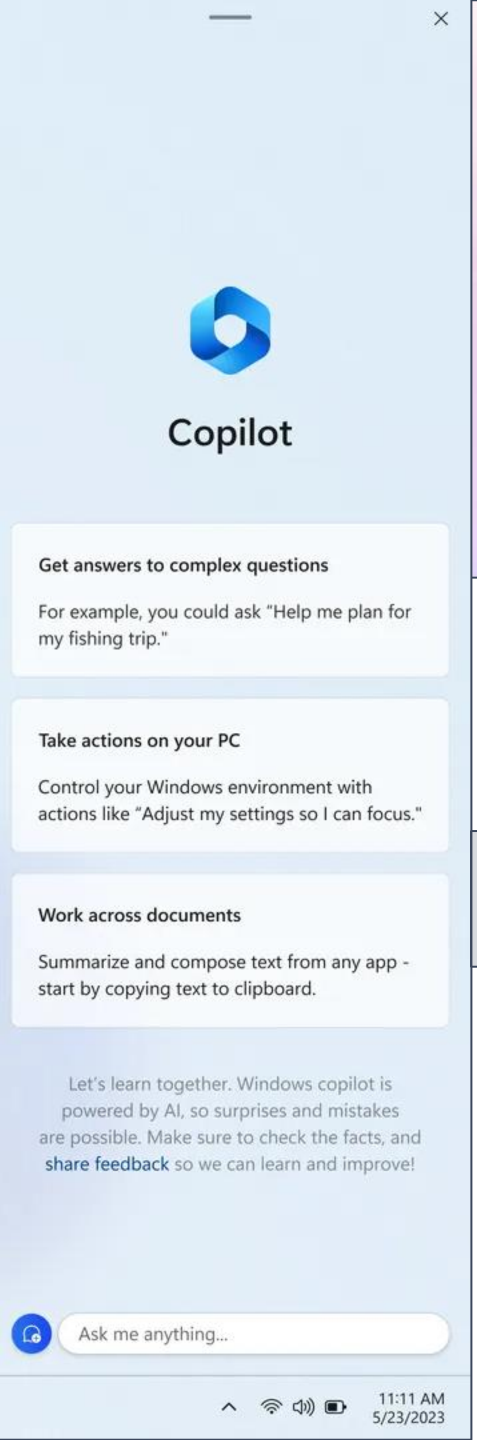
Principal Engineer @ Infobip

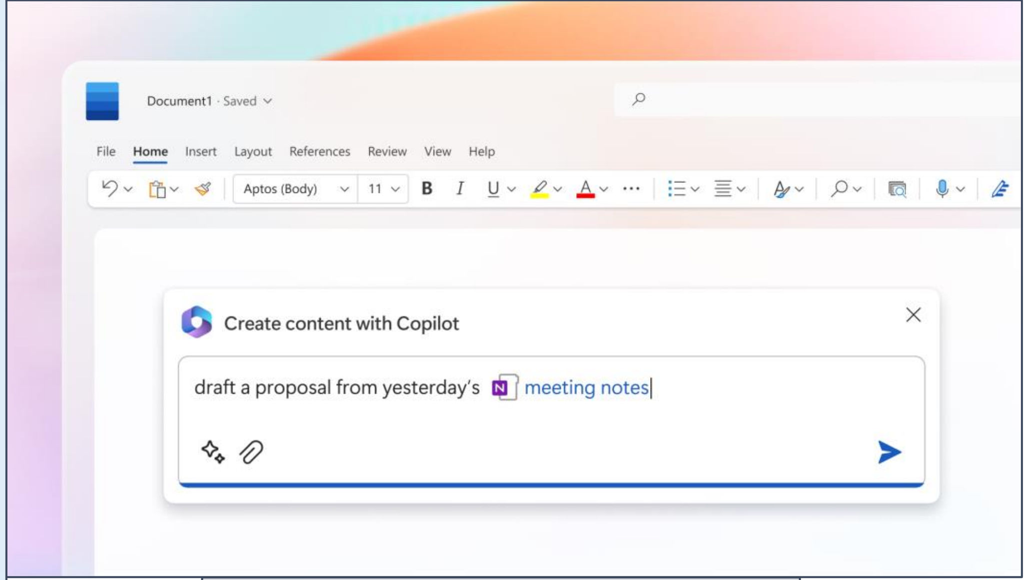


Companies everywhere are launching **copilots**

AI assistants that leverage **LLMs** to help solving a specific task

Source: https://austinhenley.com/blog/copilotpainpoints.html

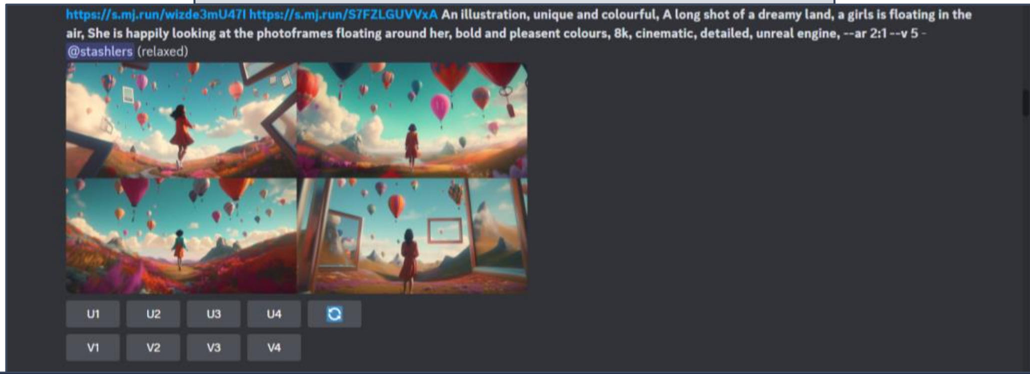


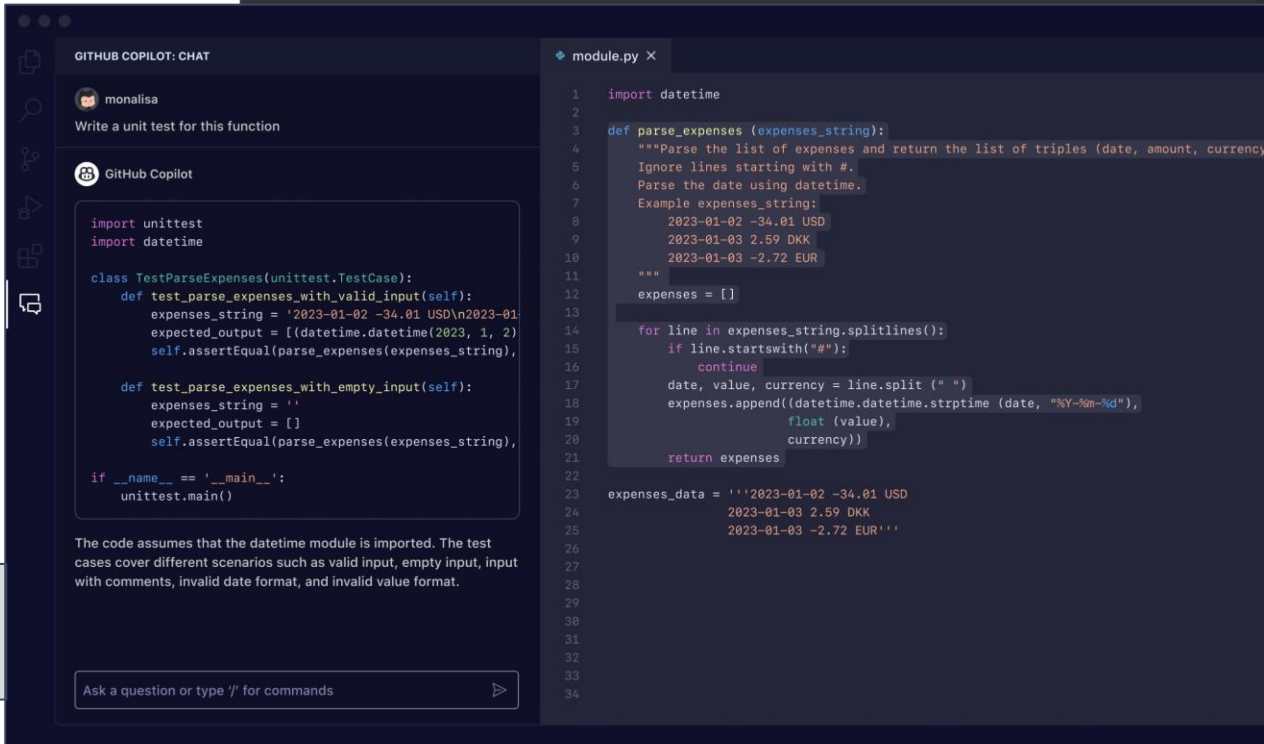


MS Office Copilot

Windows Copilot

Midjourney

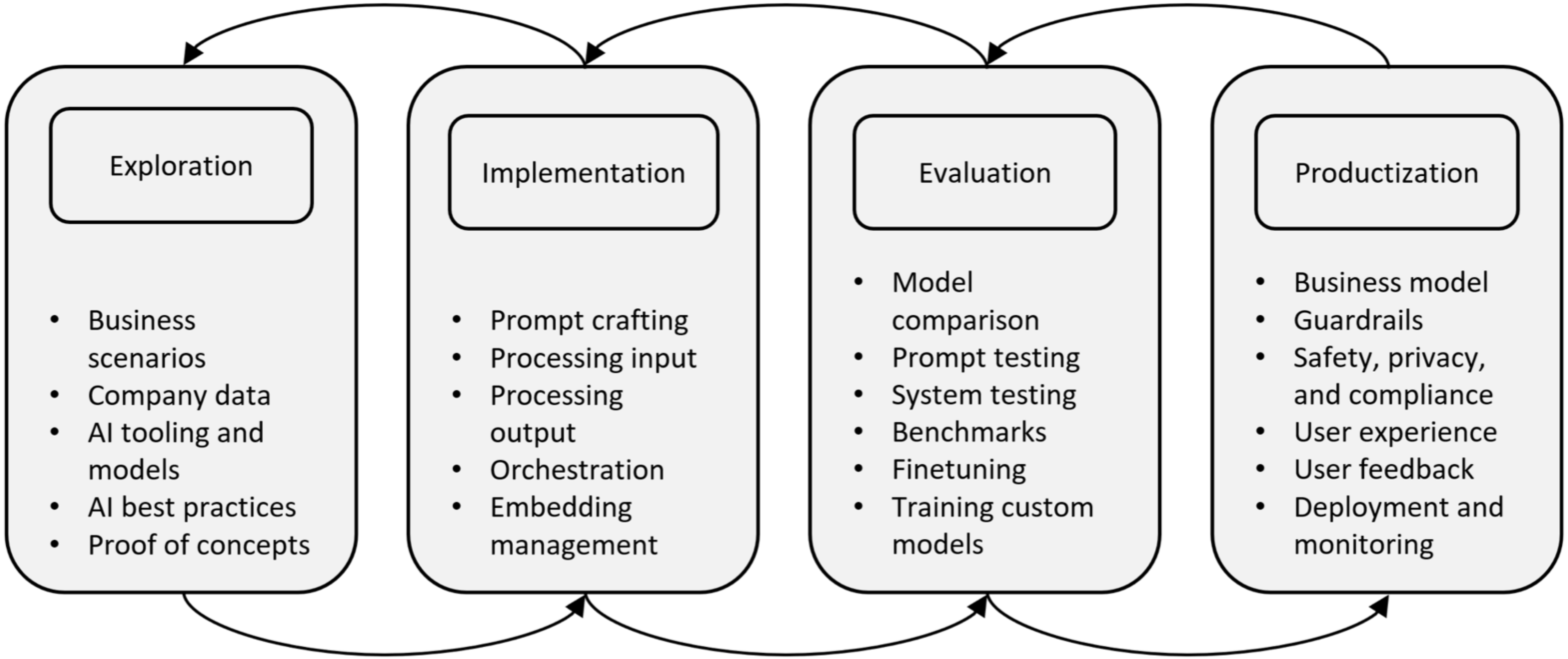


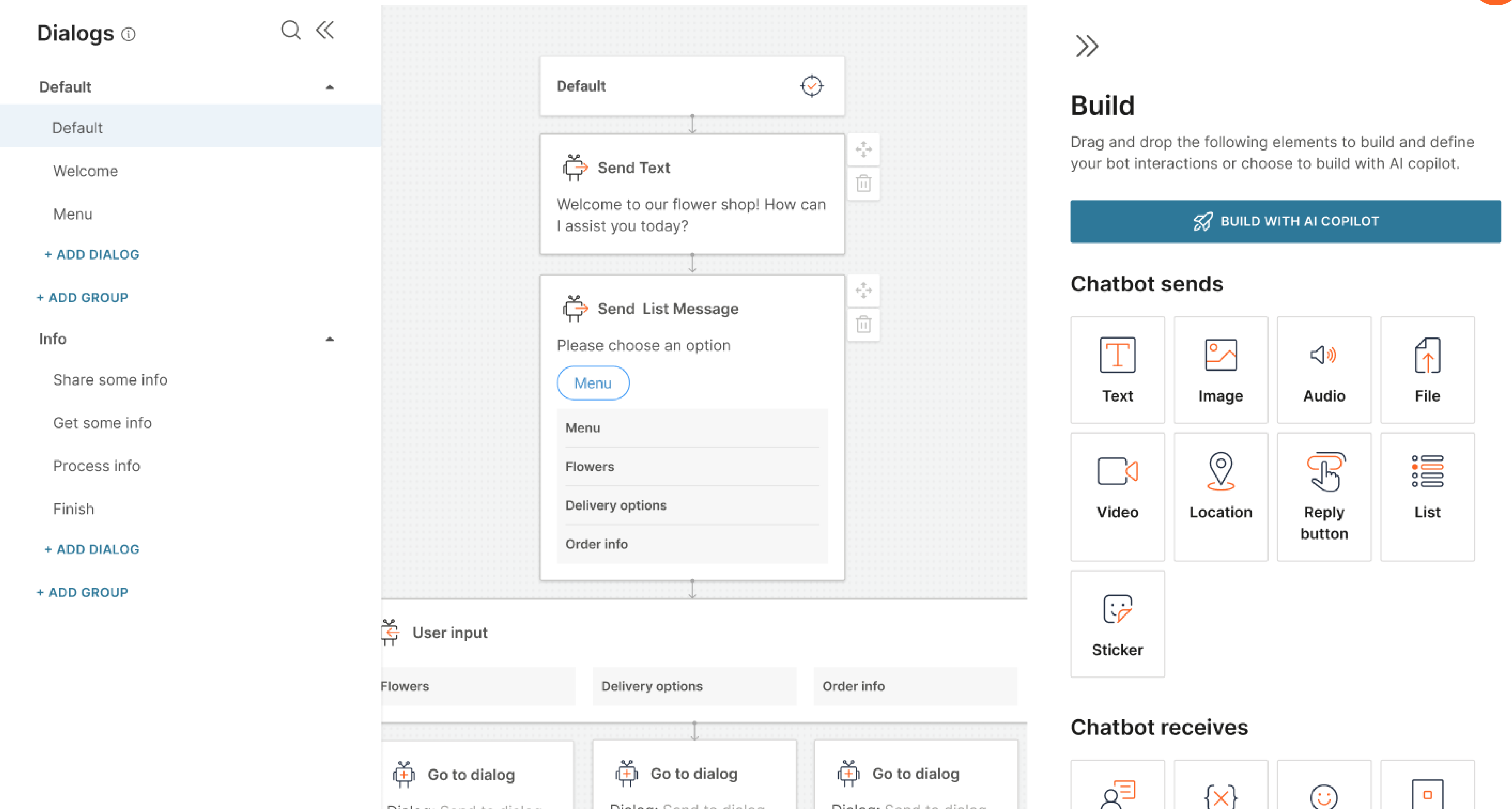


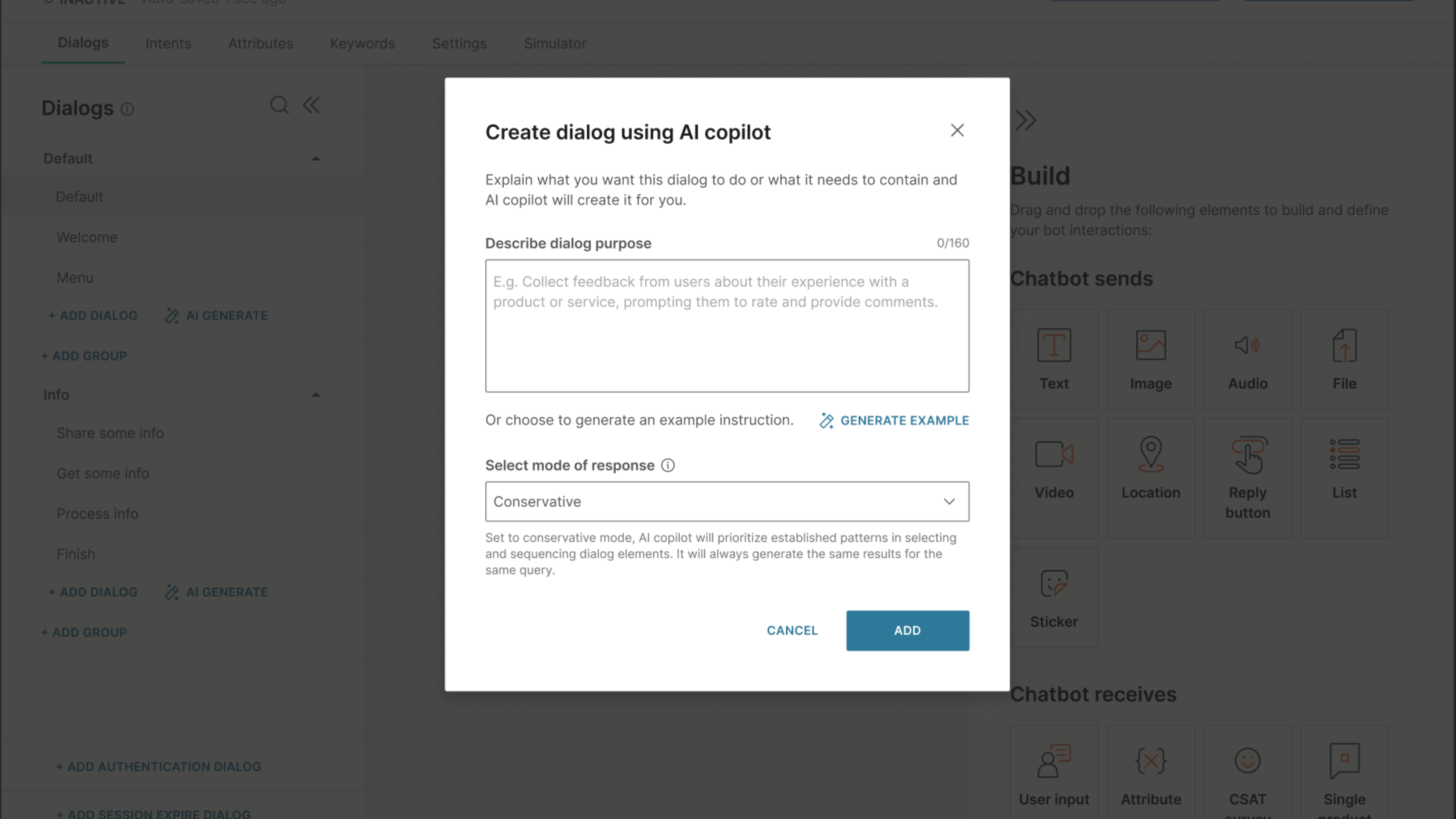
Github Copilot

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

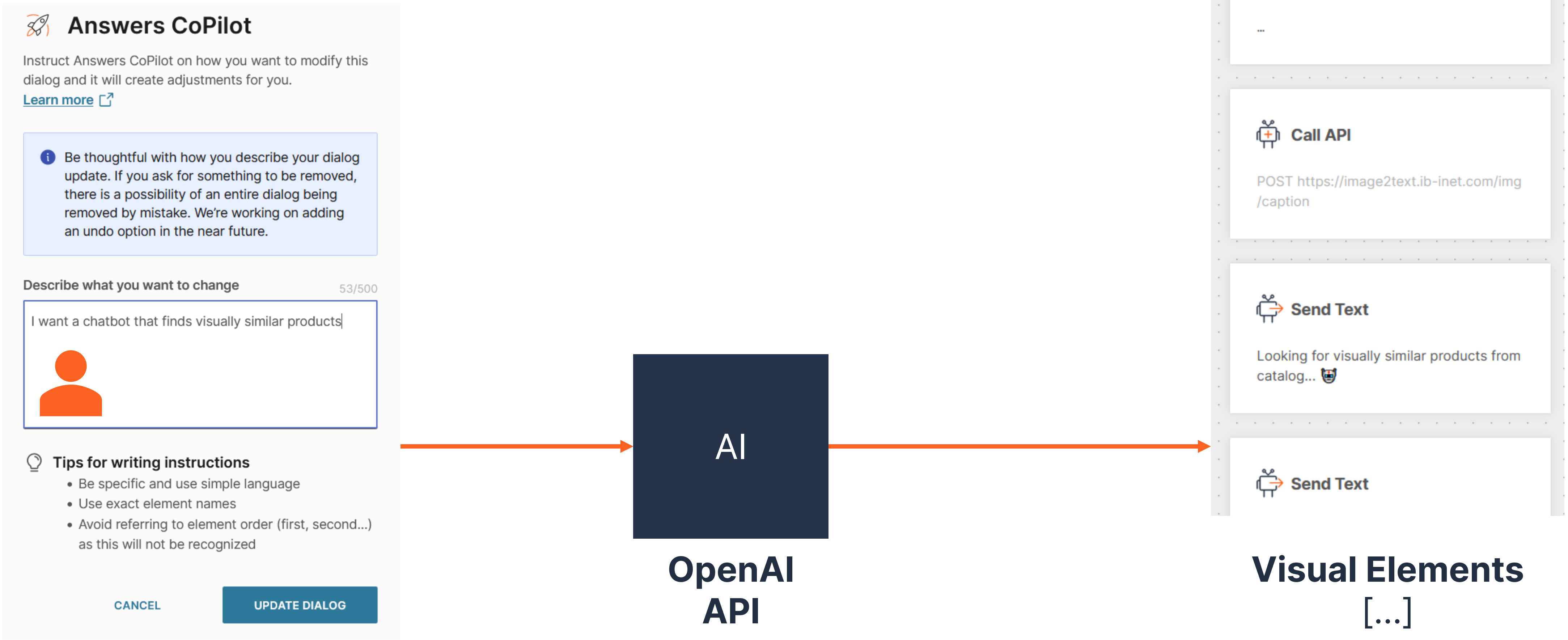
Chris Parnin, Gustavo Soares, Rahul Pandita, Sumit Gulwani, Jessica Rich, Austin Z. Henley
{chrisparnin,gustavo.soares}@microsoft.com,rahulpandita@github.com,{sumitg,jessrich,austinhenley}@microsoft.com
Microsoft, GitHub Inc.
USA




Dialogs
Default
Default
Welcome
Menu
+ ADD DIALOG
+ ADD GROUP
Info
Share some info
Get some info
Process info
Finish
+ ADD DIALOG
+ ADD GROUP
Default
Send Text
Welcome to our flower shop! How can I assist you today?
Send List Message
Please choose an option
Menu
Menu
Flowers
Delivery options
Order info
User input
Flowers
Delivery options
Order info
Go to dialog
Go to dialog
Go to dialog
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


Dialogs
Intents
Attributes
Keywords
Settings
Simulator
Dialogs
Default
Default
Welcome
Menu
+ ADD DIALOG
AI GENERATE
+ ADD GROUP
Info
Share some info
Get some info
Process info
Finish
+ ADD DIALOG
AI GENERATE
+ ADD GROUP
+ ADD AUTHENTICATION DIALOG
Create dialog using AI copilot
Explain what you want this dialog to do or what it needs to contain and AI copilot will create it for you.
Describe dialog purpose
0/160
E.g. Collect feedback from users about their experience with a product or service, prompting them to rate and provide comments.
Or choose to generate an example instruction.
GENERATE EXAMPLE
Select mode of response
Conservative
Set to conservative mode, AI copilot will prioritize established patterns in selecting and sequencing dialog elements. It will always generate the same results for the same query.
CANCEL
ADD
Build
Drag and drop the following elements to build and define your bot interactions:
Chatbot sends
Text
Image
Audio
File
Video
Location
Reply button
List
Sticker
Chatbot receives
User input
Attribute
CSAT
Single

Prompting



Answers CoPilot
Instruct Answers CoPilot on how you want to modify this dialog and it will create adjustments for you.
Learn more
Be thoughtful with how you describe your dialog update. If you ask for something to be removed, there is a possibility of an entire dialog being removed by mistake. We're working on adding an undo option in the near future.
Describe what you want to change
53/500
I want a chatbot that finds visually similar products
Tips for writing instructions
Be specific and use simple language
Use exact element names
Avoid referring to element order (first, second...) as this will not be recognized
CANCEL
UPDATE DIALOG
AI
OpenAI API
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
Visual Elements
[...]

Prompting

Test out prompt engineering baselines with API from Microsoft (GPT3.5-turbo)

Strategies:

- **Zero-Shot**
  - A prompt that describes the problem of building a chatbot dialog as well as states the vocabulary of the available visual elements
- **Few-Shot**
  - Add multiple examples of input task descriptions and their expected outputs
- **Few-Shot with Instructions**
  - Add the information about specific rules that need to be enforced to render the generated output in the UI

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

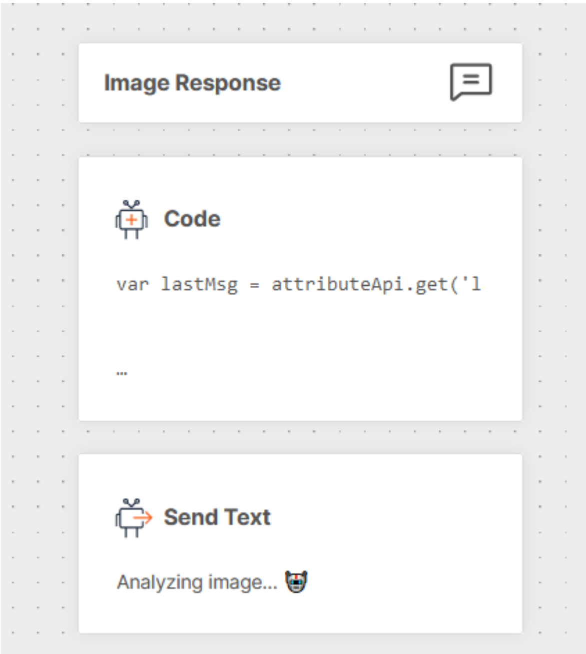
	Hallucinations	HitRate	
Zero-Shot	30.67 %	1.31 %	T = 0.0
Few-Shot	20.22 %	**2.13 %**	
Few-Shot with Instructions	23.57 %	0.68 %	
Zero-Shot	46.44 %	2.09 %	T = 0.7
Few-Shot	**12.63 %**	1.75 %	
Few-Shot with Instructions	25.70 %	0.69 %	


CAN WE
DO BETTER?

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

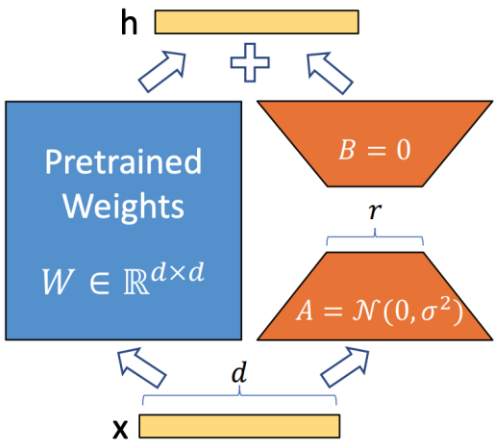
Hypothesis: You can use **GenAI** (e.g., GPT3.5-turbo) to **synthetically** create **description data**

```
import json

instruction = """
You are a chabtot generator. Your job is to find out and desribe 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

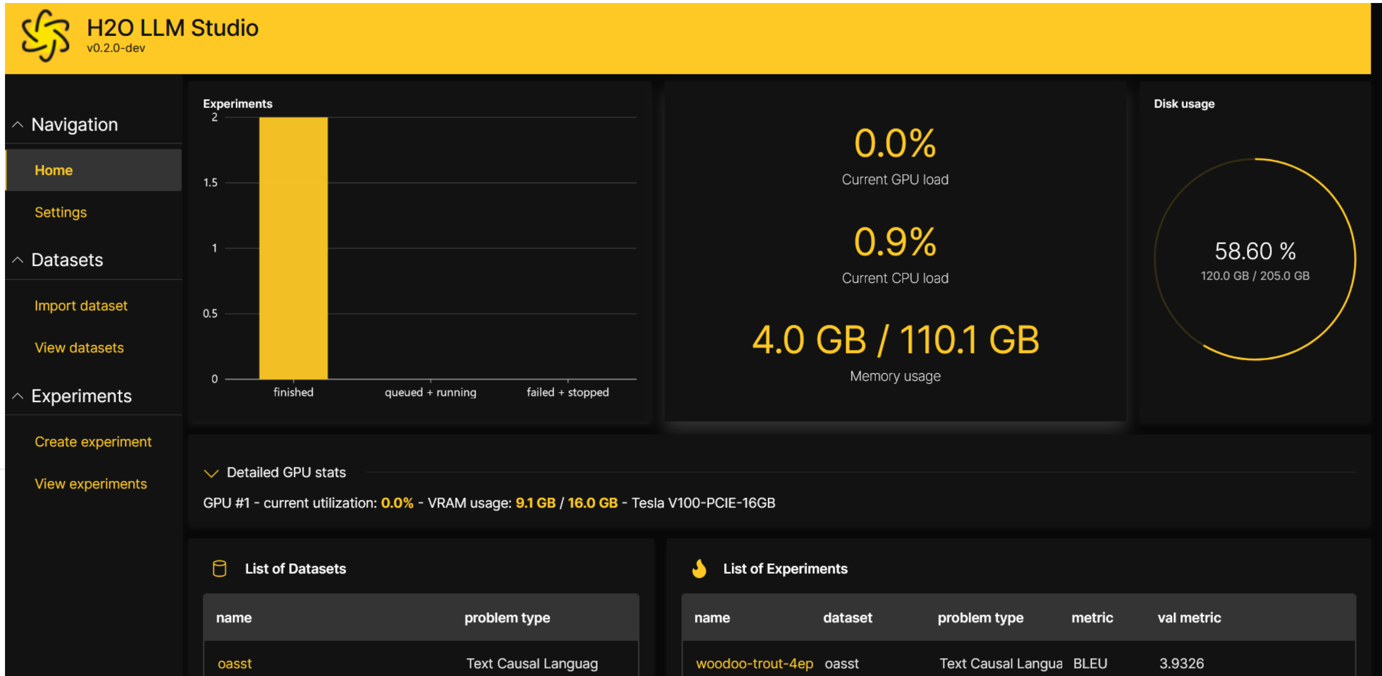


PEFT

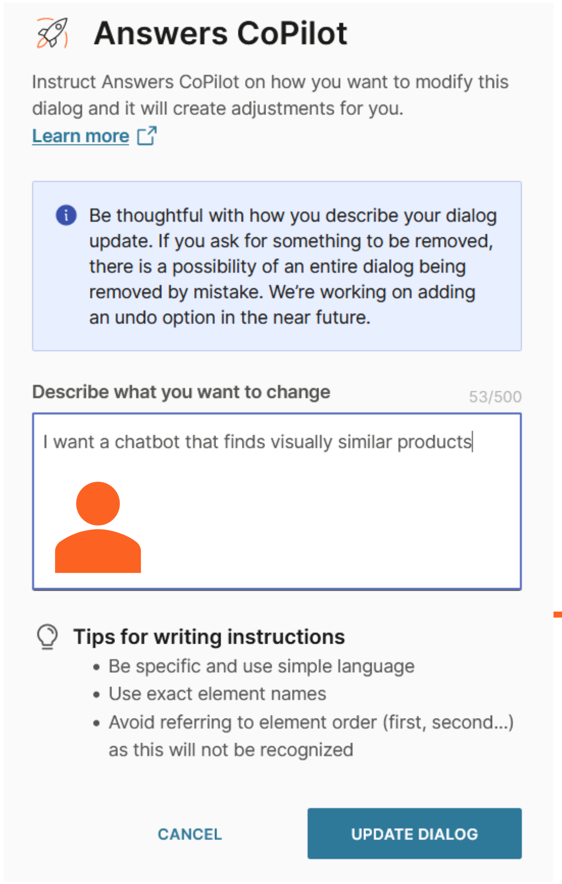
State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods

https://github.com/huggingface/peft

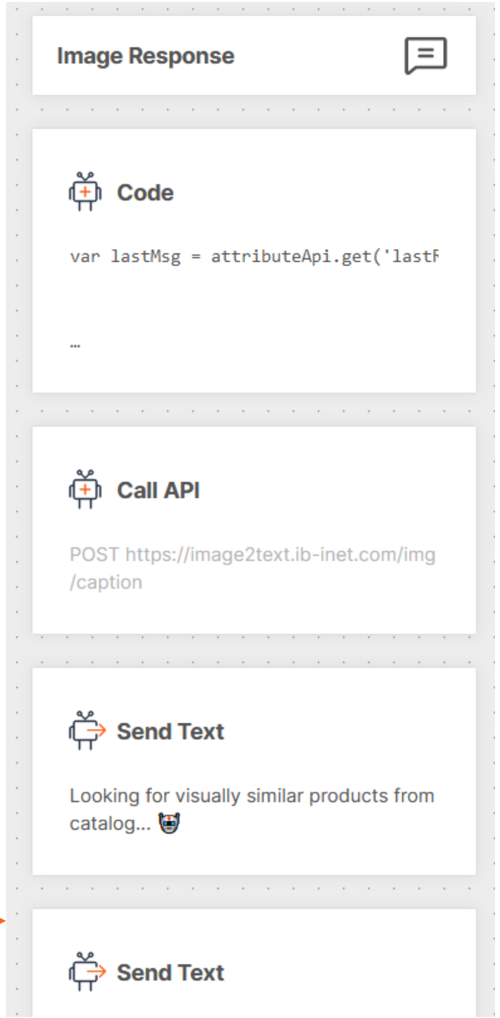
https://github.com/h2oai/h2o-llmstudio



Fine-Tuned Models



LLMs fine-tuned on relevant data



Visual Elements
[...]

Fine-Tuned Models

Fine-tuned LLMs were able to achieve

- Number of **Hallucinations** significantly lowered from 12.63% → the **best performance of 0.04%**
- A **HitRate** that improved from 0.68% - 2.13% → **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 %

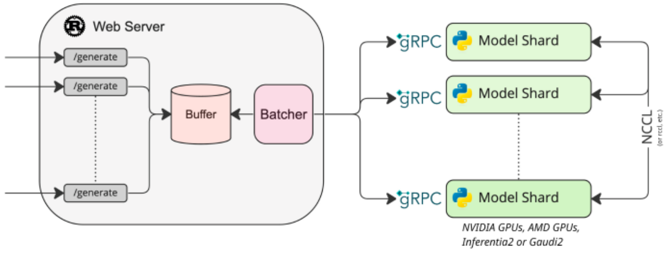
Inference

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

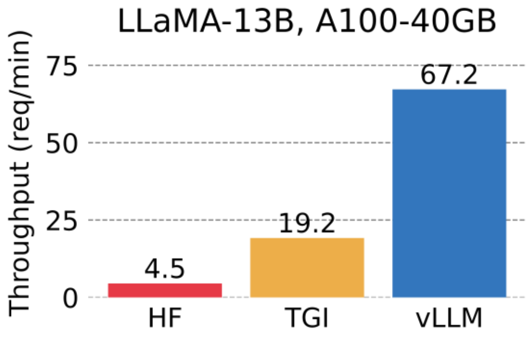



```
docker run --detach --gpus all --shm-size 1g -p 9999:80 -v /var/lib/docker/volumes/h2o-llmstudio-shared/output/user:/data
  ghcr.io/huggingface/text-generation-inference:1.1.0  --model-id /data/mymodel
```

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

- https://github.com/ggerganov/llama.cpp

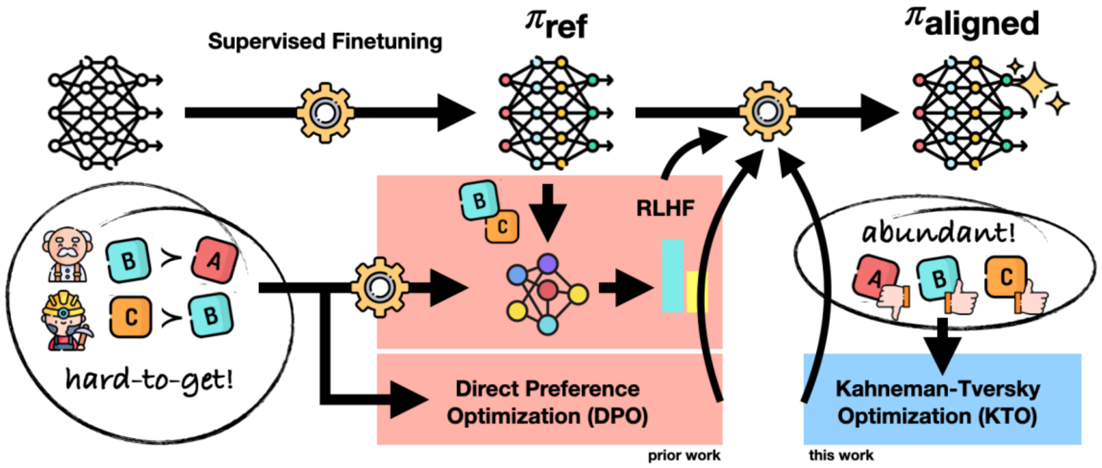
	VRAM
Sheared LLaMA 1.3B **(tiny)**	5.1 GB
LLaMa 3B **(small)**	9.5 GB
Mistral 7B Instruct **(mid)**	13.6 GB



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


NEXT
STEPS

Human-Aware Loss Functions



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!

https://github.com/ContextualAI/HALOs

KTO: Model Alignment as Prospect Theoretic Optimization

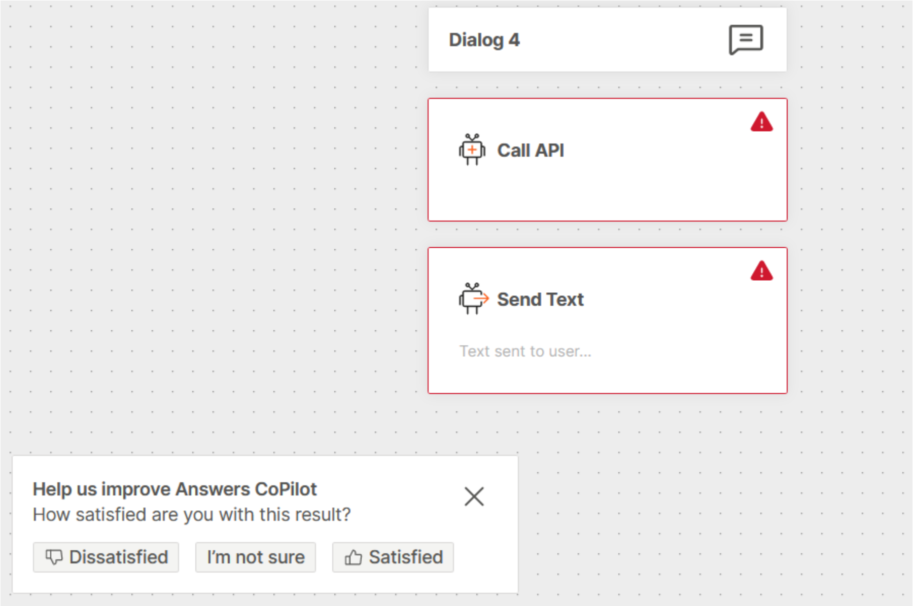
Kawin Ethayarajh[1] Winnie Xu[2] Niklas Muennighoff[2] Dan Jurafsky[1] Douwe Kiela[1 2]

Abstract

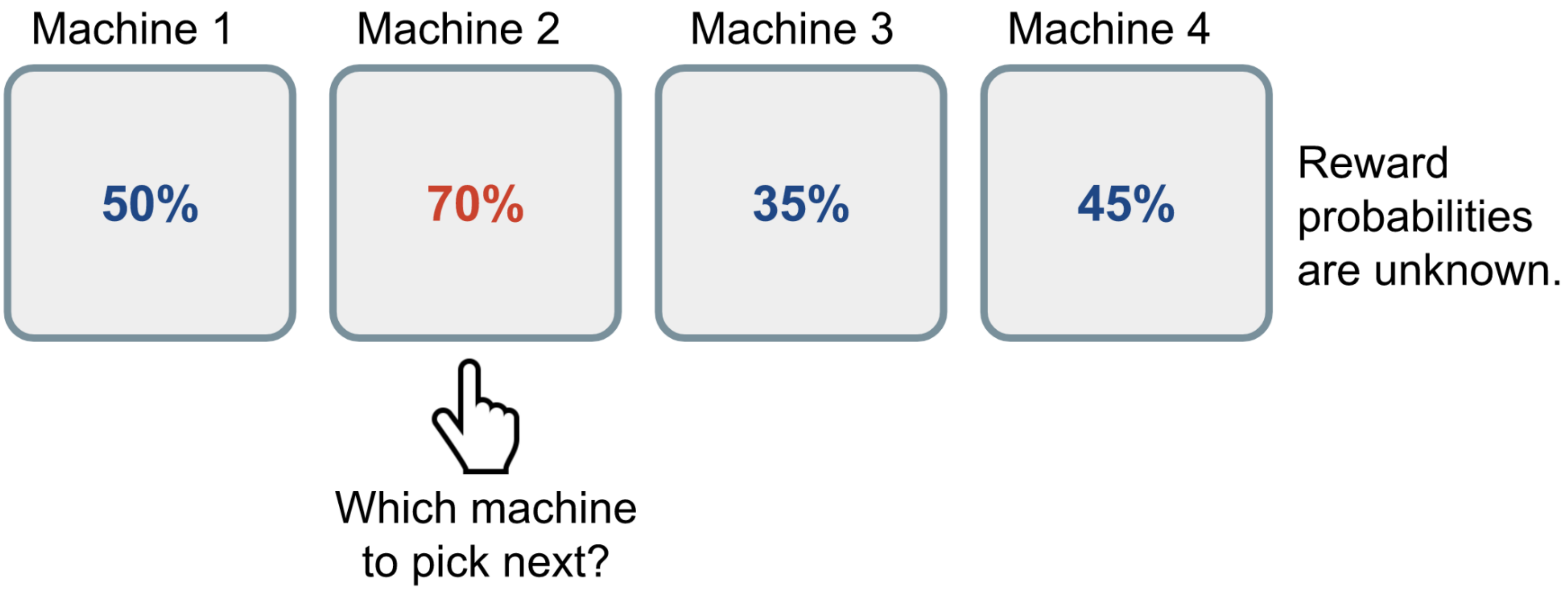
Kahneman & Tversky's *prospect theory* tells us that humans perceive random variables in a biased but well-defined manner (1992); for example, humans are famously loss-averse. We show that objectives for aligning LLMs with human feedback implicitly incorporate many of these biases—the success of these 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



Multi-Armed Bandits



Source: https://lilianweng.github.io/posts/2018-01-23-multi-armed-bandit/

Multi-Armed Bandits

Let :

- $\Pi_0 = (\pi_1^0, \ldots, \pi_K^0)$ be a prior distribution over $(\theta_1, ..., \theta_K)$
- $\Lambda_t = (\lambda_1^t, \ldots, \lambda_K^t)$ be the posterior over the means $(\mu_1, ..., \mu_K)$ a the end of round t

The **Bayes-UCB algorithm** chooses at time t

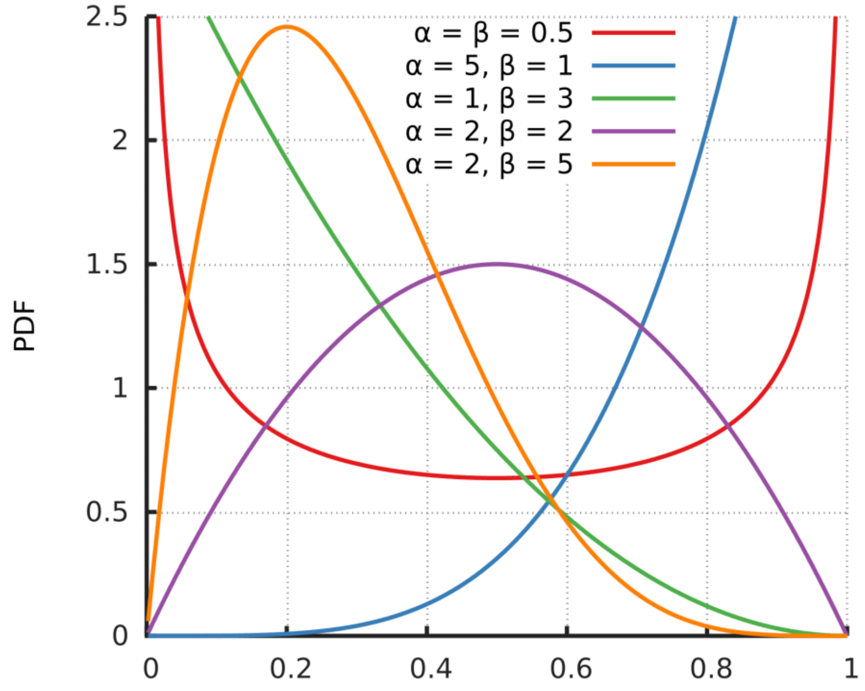
$$A_t = \underset{a}{\text{argmax}}\ Q\left(1 - \frac{1}{t(\log t)^c}, \lambda_a^{t-1}\right)$$

where $Q(\alpha, \pi)$ is the quantile of order α of the distribution π.

Bernoulli reward with uniform prior: $\theta = \mu$ and $\Pi_t = \Lambda_t$

$$A_t = \underset{a}{\text{argmax}}\ Q\left(1 - \frac{1}{t(\log t)^c}, \text{Beta}(S_a(t) + 1, N_a(t) - S_a(t) + 1)\right)$$

Kaufmann, E. Bayesian and Frequentist Methods in Bandit Models. 2013



Multi-Armed Bandits

Let :

- $\Pi_0 = (\pi_1^0, \ldots, \pi_K^0)$ be a prior distribution over $(\theta_1, ..., \theta_K)$
- $\Lambda_t = (\lambda_1^t, \ldots, \lambda_K^t)$ be the posterior over the means $(\mu_1, ..., \mu_K)$ a the end of round t

The **Bayes-UCB algorithm** chooses at time t

$$A_t = \underset{a}{\text{argmax}}\ Q\left(1 - \frac{1}{t(\log t)^c}, \lambda_a^{t-1}\right)$$

where $Q(\alpha, \pi)$ is the quantile of order α of the distribution π.

Bernoulli reward with uniform prior: $\theta = \mu$ and $\Pi_t = \Lambda_t$

$$A_t = \underset{a}{\text{argmax}}\ Q\left(1 - \frac{1}{t(\log t)^c}, \text{Beta}(S_a(t) + 1, N_a(t) - S_a(t) + 1)\right)$$

Kaufmann, E. Bayesian and Frequentist Methods in Bandit Models. 2013

```python
class BayesianUCBBandit:
    def __init__(self, n_arms):
        self.n_arms = n_arms
        self.alpha = np.ones(n_arms)
        self.beta = np.ones(n_arms)
        self.t = 0
        self.c = 5 # exploration parameter
        self.totals = np.zeros(n_arms, dtype=int)

    def select_arm(self):
        self.t += 1

        quantile_level = 1 - 1 / self.t
        # avoid division by 0 when t = 1
        if self.t > 1:
            quantile_level = 1 - 1 / (self.t * (np.log(self.t) ** self.c))

        ucb_values = [beta.ppf(quantile_level, a, b) for a, b in zip(self.alpha, self.beta)]

        return np.argmax(ucb_values)

    def update(self, chosen_arm_index, feedback):
        if feedback == 1:
            self.alpha[chosen_arm_index] += 1  # Positive
        elif feedback == -1:
            self.beta[chosen_arm_index] += 1   # Negative
        # Else, do nothing (no feedback was given)
        # we could also have different weights for feedback
```

Multi-Armed Bandits

```python
chosen_LLM_index = bandit.select_arm()

output = generate(input_text, chosen_LLM_index)

# feedback aquisition needs to be defined
feedback = get_feedback(chosen_LLM_index, input_text, output)

bandit.update(chosen_LLM_index, feedback)
```

```python
class BayesianUCBBandit:
    def __init__(self, n_arms):
        self.n_arms = n_arms
        self.alpha = np.ones(n_arms)
        self.beta = np.ones(n_arms)
        self.t = 0
        self.c = 5 # exploration parameter
        self.totals = np.zeros(n_arms, dtype=int)

    def select_arm(self):
        self.t += 1

        quantile_level = 1 - 1 / self.t
        # avoid division by 0 when t = 1
        if self.t > 1:
            quantile_level = 1 - 1 / (self.t * (np.log(self.t) ** self.c))

        ucb_values = [beta.ppf(quantile_level, a, b) for a, b in zip(self.alpha, self.beta)]

        return np.argmax(ucb_values)

    def update(self, chosen_arm_index, feedback):
        if feedback == 1:
            self.alpha[chosen_arm_index] += 1  # Positive
        elif feedback == -1:
            self.beta[chosen_arm_index] += 1   # Negative
        # Else, do nothing (no feedback was given)
        # we could also have different weights for feedback
```

infobip

Questions?

Emanuel Lacić
Principal Engineer @ Infobip

emanuel.lacic@infobip.com
@elacic1
/in/elacic
http://elacic.me