Durham University

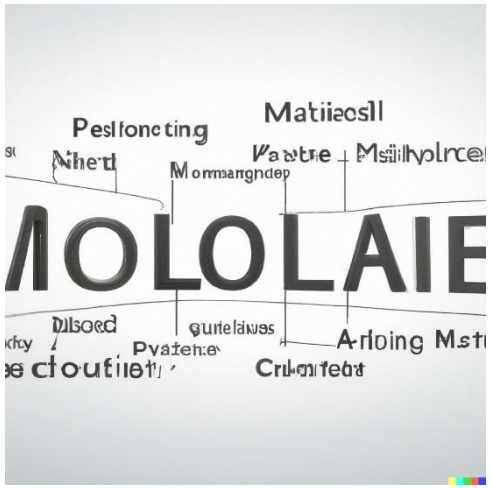
COMP 3647
Human-AI Interaction Design

Topic 6
Large Language Model Applications (LLMAs)

Prof. Effie L-C Law

Large Language Models (LLMs): Use Cases & Applications

- Generative
- Summarization
- Rewrite
- Search
- Question Answering
- Clustering
- Classification



Durham University

DALL.E: Large language models human-AI interaction design

Large Language Models (LLMs): Generative

- GPT-3 (and ChatGPT), LaMDA, Character.ai, Megatron-Turing NLG – Text generation useful especially for dialogue with humans, as well as copywriting, translation, and other tasks
- PaLM – LLM from Google Research that provides several other natural language tasks
- Anthropic.ai – Product focused on optimizing the sales process, via chatbots and other LLM-powered tools
- BLOOM – General purpose language model used for generation and other text-based tasks, and focused specifically on multi-language support
- Codex (and Copilot), CodeGen – Code generation tools that provide auto-complete suggestions as well as creation of entire code blocks
- DALL-E, Stable Diffusion, MidJourney – Generation of images based on text descriptions
- Imagen Video – Generation of videos based on text descriptions
- Whisper – Transcription of audio files into text

Source: Vectara.com

Large Language Models (LLMs): Summarisation

- Assembly AI – provides transcription and summarization of audio and video
- Davinci – a GPT-3 based model that can summarize text, among several other tasks
- Cohere Generate – LLM-based product that can paraphrase text and distill long passages down to condense points
- Megatron-Turing NLG – LLM that can perform a broad set of natural language tasks, including summarization
- Viable – summarizes data spread out across multiple sources to improve business operations and efficiency

Source: Vectara.com

Large Language Models (LLMs): Rewrite

- Grammarly – Grammatical error correction tool
- Cohere Generate – LLM-based product that can rewrite text, for example to clean it up or change the voice
- Google Translate – translates over 100 languages
- Meta AI's NLLB-200 – translates over 200 languages

Source: Vectara.com

Large Language Models (LLMs): Search

- Vectara – LLM-powered search platform which matches data based on intent and meaning, regardless of how the concepts are worded.
- Glean – workplace search that helps you find information across your company's applications
- Neeva – search engine providing ad-free results over data crawled from the Internet, with the option to also data in your personal accounts searchable
- Azure Embeddings Models, OpenAI Embeddings Models – these generate text embeddings that can be used as the basis for a custom-built search system
- Jina – neural search platform that provide prompt optimization and decision support capabilities
- You.com – search engine that leverages LLMs to help make users' search activities more efficient

Source: Vectara.com

Large Language Models (LLMs): Question Answering

- Google Search, Bing Search – both of these regularly attempt to provide a summarized answer at the top of a list of search results

- LLaMA – focused especially on question answering and document summarization

- Vectara – retrieval of relevant information based on the user's query/prompt, which is then summarized to provide an answer with citations

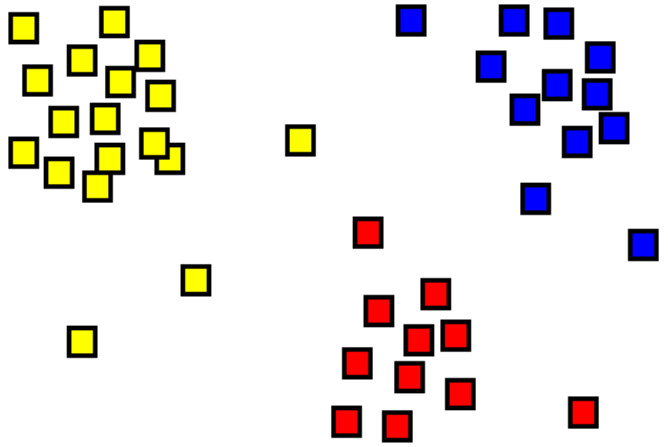
- Neeva – in addition to search results (as mentioned above) summarized answers are provided to the user

- Contriever – LLM from Facebook Research that has been trained for information retrieval and question answering

Source: Vectara.com

Large Language Models (LLMs): Clustering

Cohere Embed, Azure Embeddings Models, OpenAI Embeddings Models – these generate text embeddings that can be used as the basis for a custom-built clustering application



Source: Vectara.com

Categorisation of LLM Usage: 1-3

Usage	Example Applications
1. Natural Language Processing (NLP) Tasks	▪ **Tools:** ChatGPT, Google BERT, Hugging Face Transformers, SpaCy ▪ **Sentiment Analysis** ▪ **Translation:** Google Translate, DeepL ▪ **Grammar and style correction**: Grammarly ▪ **Search & Information retrieval**: Elasticsearch, Apache Solr ▪ **Automated content moderation**: Hate speech detection
2. Content Generation	▪ **Video game quest:** AI Dungeon (an interactive text-based game) ▪ **Code generation and debugging**: CoPilot; ▪ **Music composition**: Jukedeck ▪ **Creative writing**: songs, scripts, blogs
3. Customer service and support	▪ **Zendesk Answer Bot** ▪ **LivePerson Conversational AI**

Categorisation of LLM Usage: 4-6

Usage	Example Applications
4. Virtual Personal Assistant	▪ **Home automation**: Controlling smart home devices through voice commands with the help of virtual assistants (Siri, Alexa, Google Home) ▪ **Travel planning**: Assisting users with itinerary planning, flight bookings, and hotel reservations (Booking.com AI Trip Planner)
5. Education	▪ **Language learning tutors**: Personalized language learning experiences with practice and feedback (e.g. Duolingo) ▪ **Auto-grading**: Grading and providing feedback on student essays and assignments. ▪ **Course recommendations**: Coursera.
6. Healthcare	▪ **Medical image analysis**: Assisting in the analysis and diagnosis of medical images like X-rays and MRIs. ▪ **Virtual Health Assistants**: Providing personalized health advice and medication reminders to patients (e.g. Babylon Health)

Categorisation of LLM Usage: 7-9

Usage	Example Applications
7. Financial analysis	**Trading Bots**: Using sentiment analysis to inform trading decisions and optimize investment strategies.
8. Legal and compliance	**Contract Generation**: Automatically generating contract templates with customizable clauses and terms.
9. Academic research	▪ **Language modelling research**: Utilising large language models to explore language structures and improve AI language understanding. ▪ **Data analysis**: Processing and analysing large volumes of textual data in academic studies.

Human-AI Collaboration:

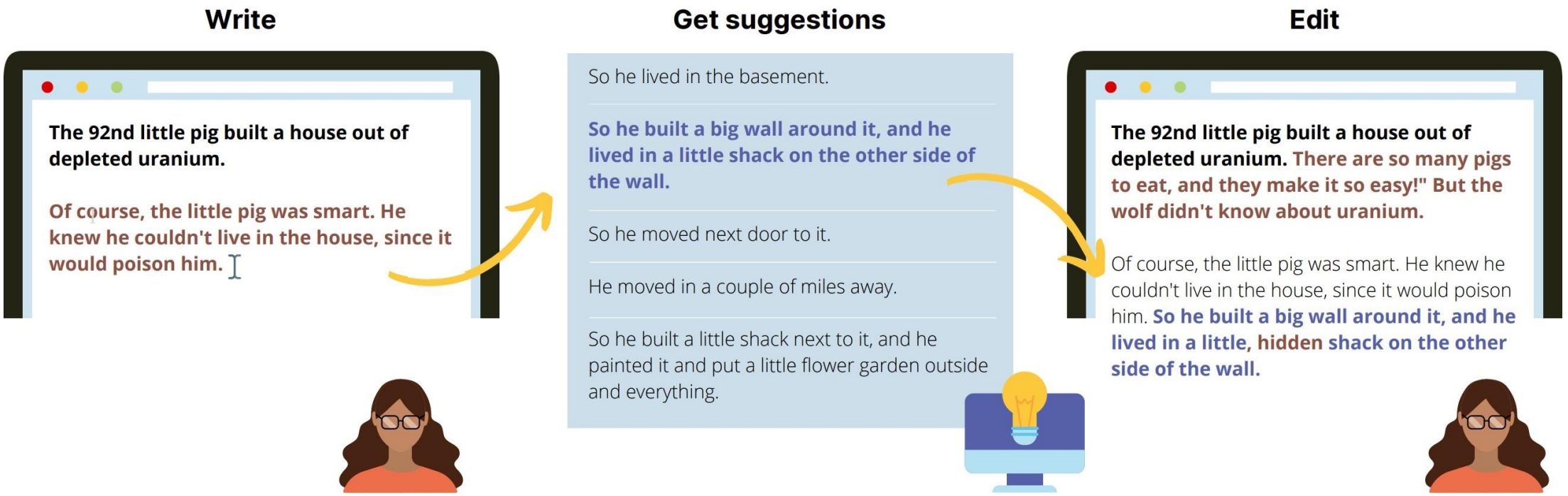
Collaborating with LLMs - Prompting

Slides credit to Sherry T. Wu (CMU) and Diyi Yang (Stanford)

Durham University

Human-AI Collaboration

The **cooperative** and **coordinated** interaction between humans (mostly non-AI experts) and AI to solve *complex problems* or *achieve certain goals*.

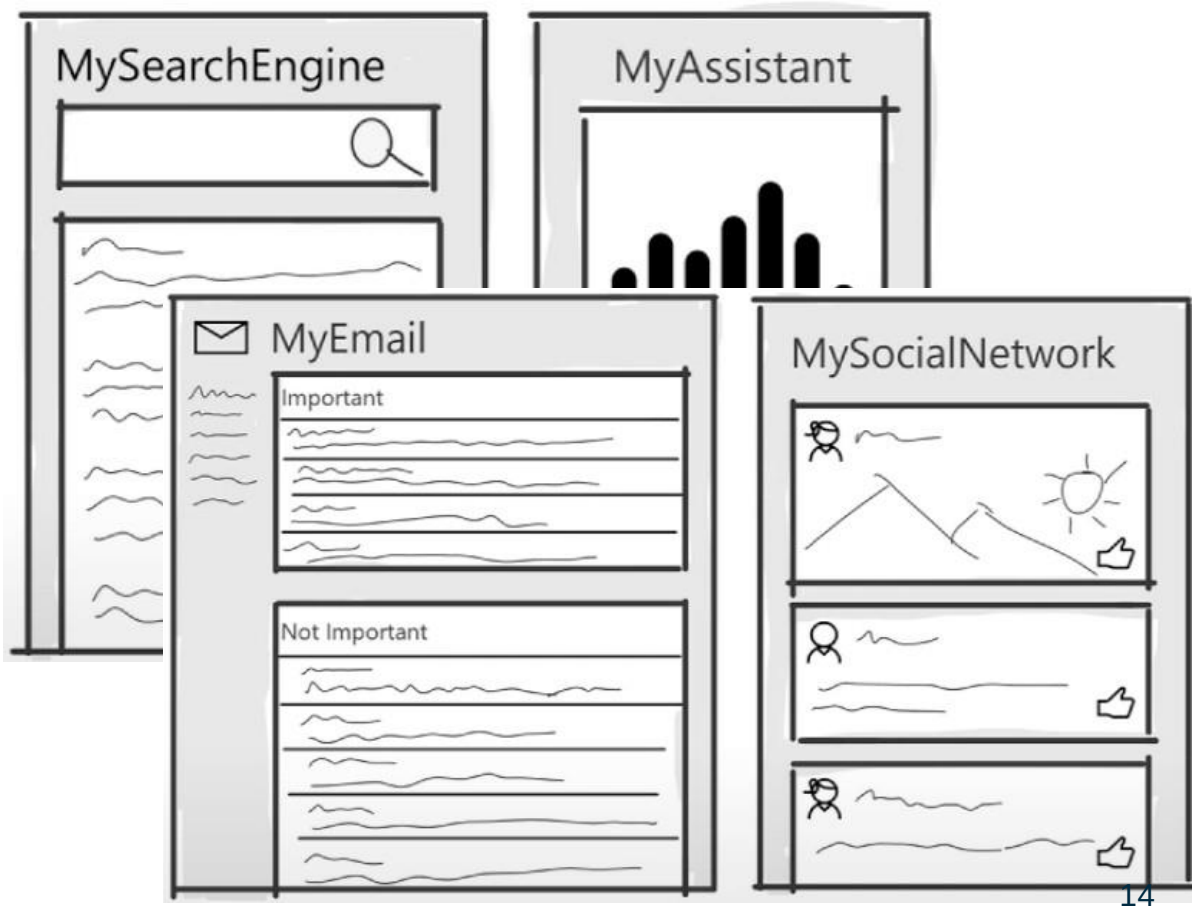


Assistance from AI-infused Applications

Similar to humans interacting with models: humans are still mostly **end users** and **domain experts**. The big difference is **AI is not a partner**, but a tool (and part of "AI-infused applications")

Because we want people to get smooth assistance from AIs when they are in the larger application context, the concept of task & AI model is blurred.

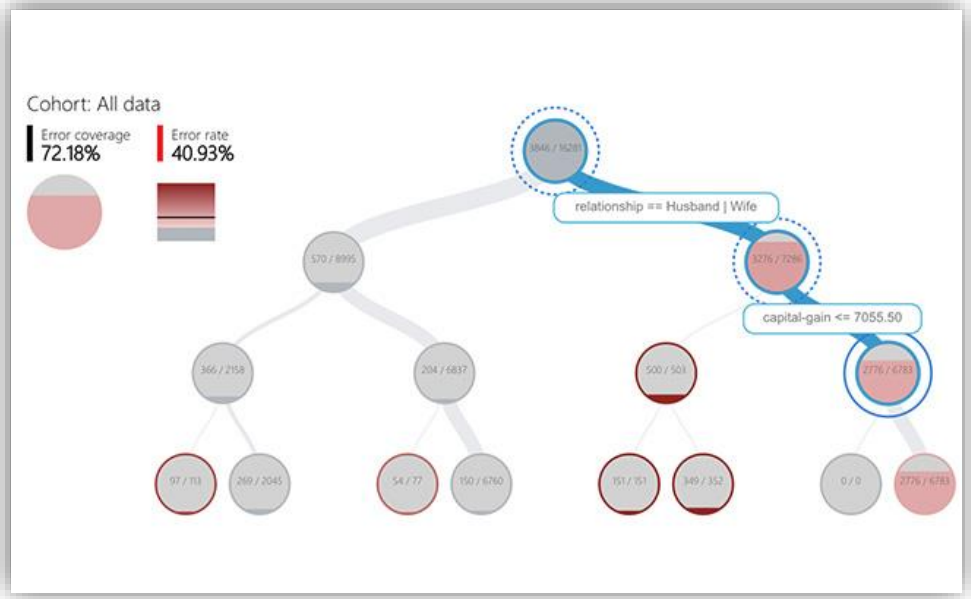
because these models are wrapped under mature visual interfaces, people tend to have less tolerance when they get wrong.

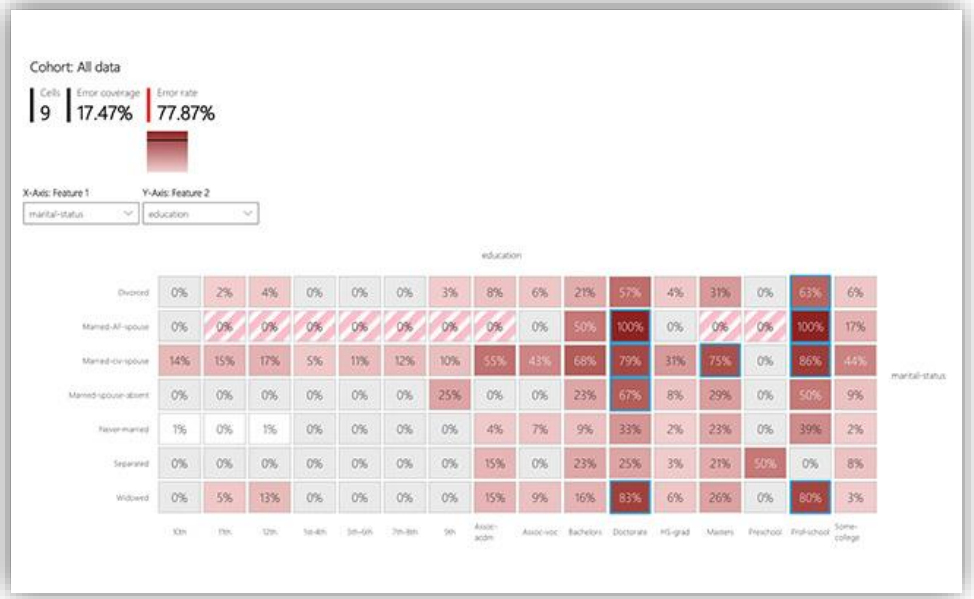


Humans Analyse Models

AI experts can systematically understand ML models and go beyond aggregated scores.

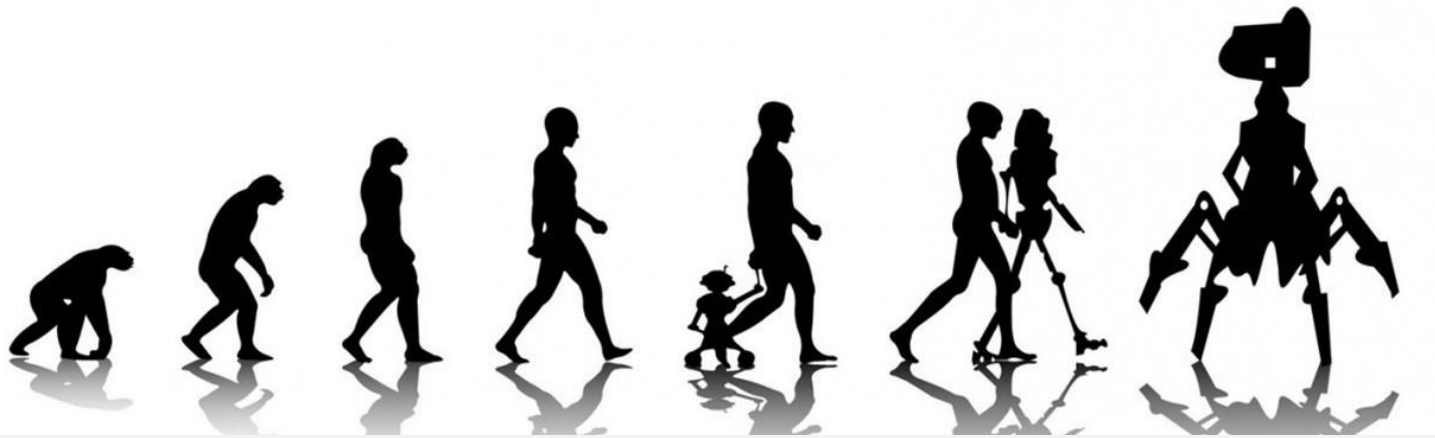
“Understanding the **broader terrain** of errors is an important starting point in pursuing systems that are robust, safe, and fair… [We need to] identify **cohorts** with higher error rates and diagnose the **root causes** behind these errors.” Eric Horvitz / Microsoft, 2021





https://erroranalysis.ai/

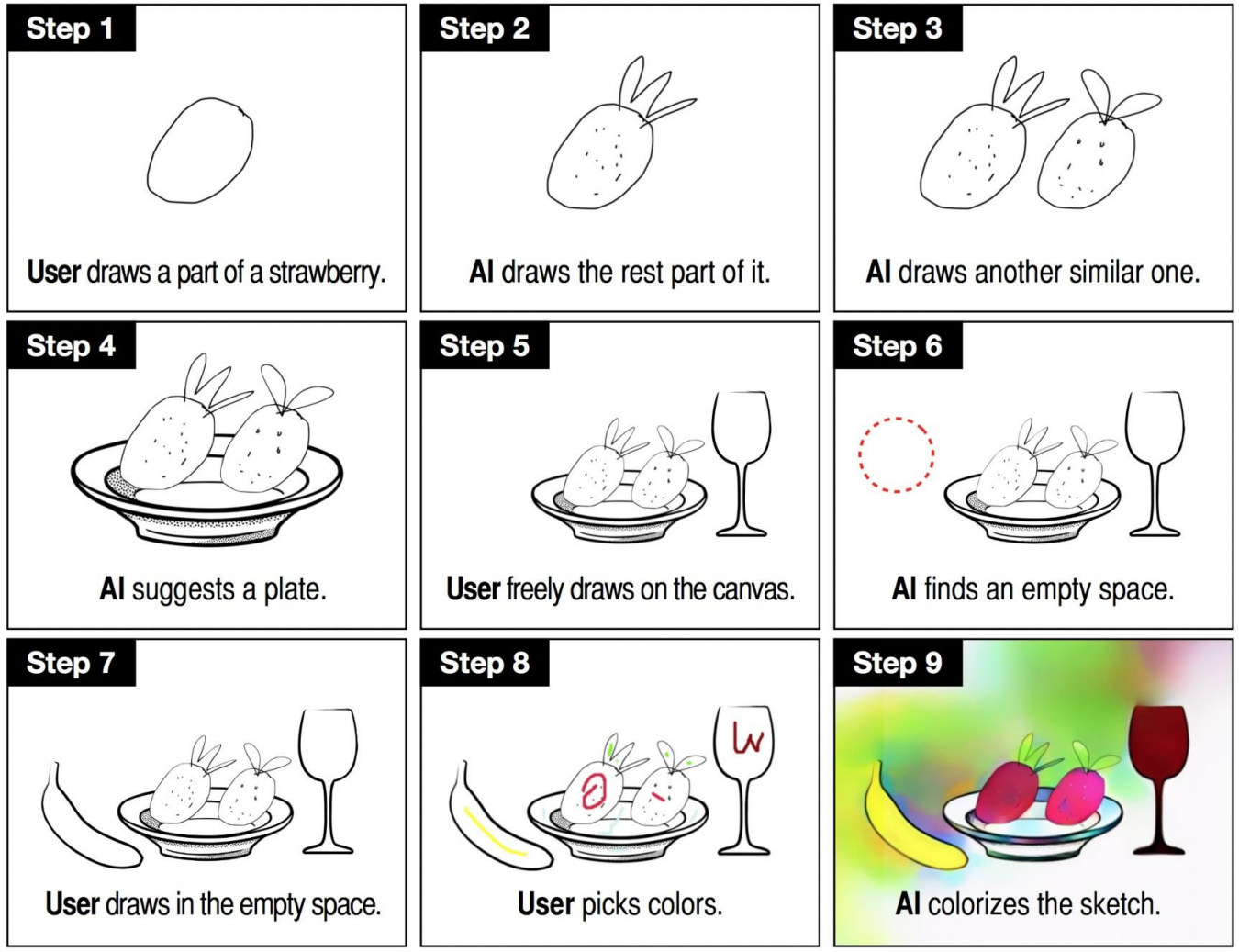
AI does not automate jobs.
It automates certain skills in jobs



AI will indeed automate most repetitive and physical tasks…and will push human professionals up the skillset ladder into uniquely human skills such as creativity, social abilities, empathy, and sense-making, which machines cannot automate.

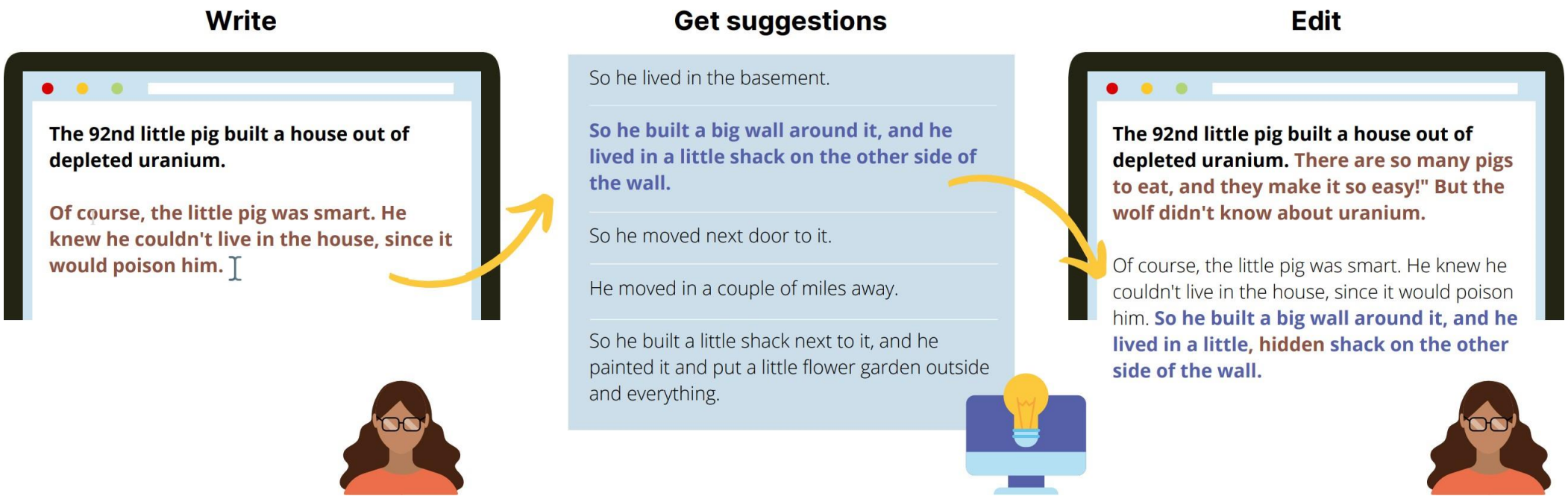
Pedro Uria-Recio, 2019

Examples of HAI Collaboration: Co-drawing



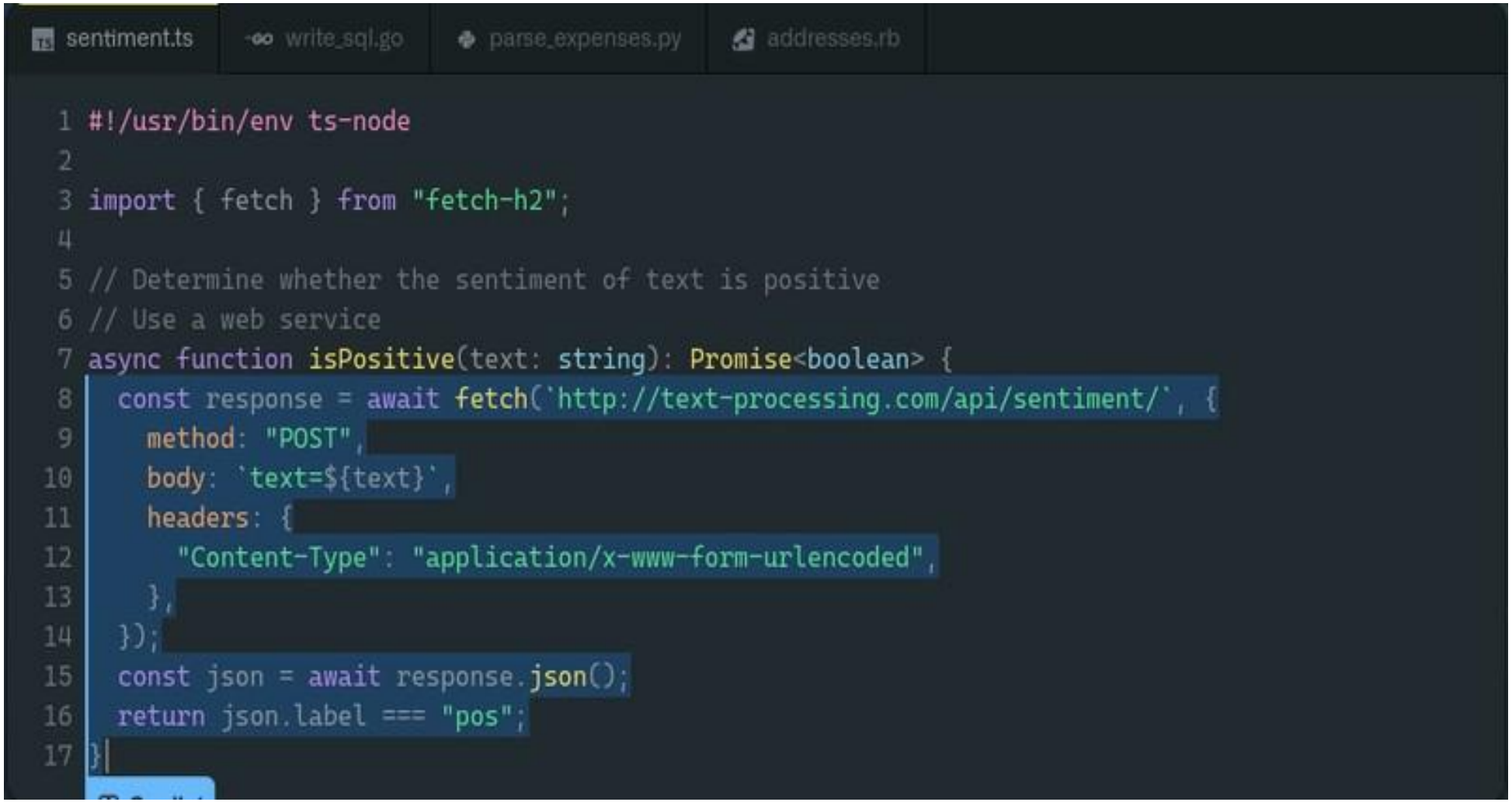
Oh, Changhoon, et al. "I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence." CHI 2018

Examples of HAI Collaboration: Co-authoring



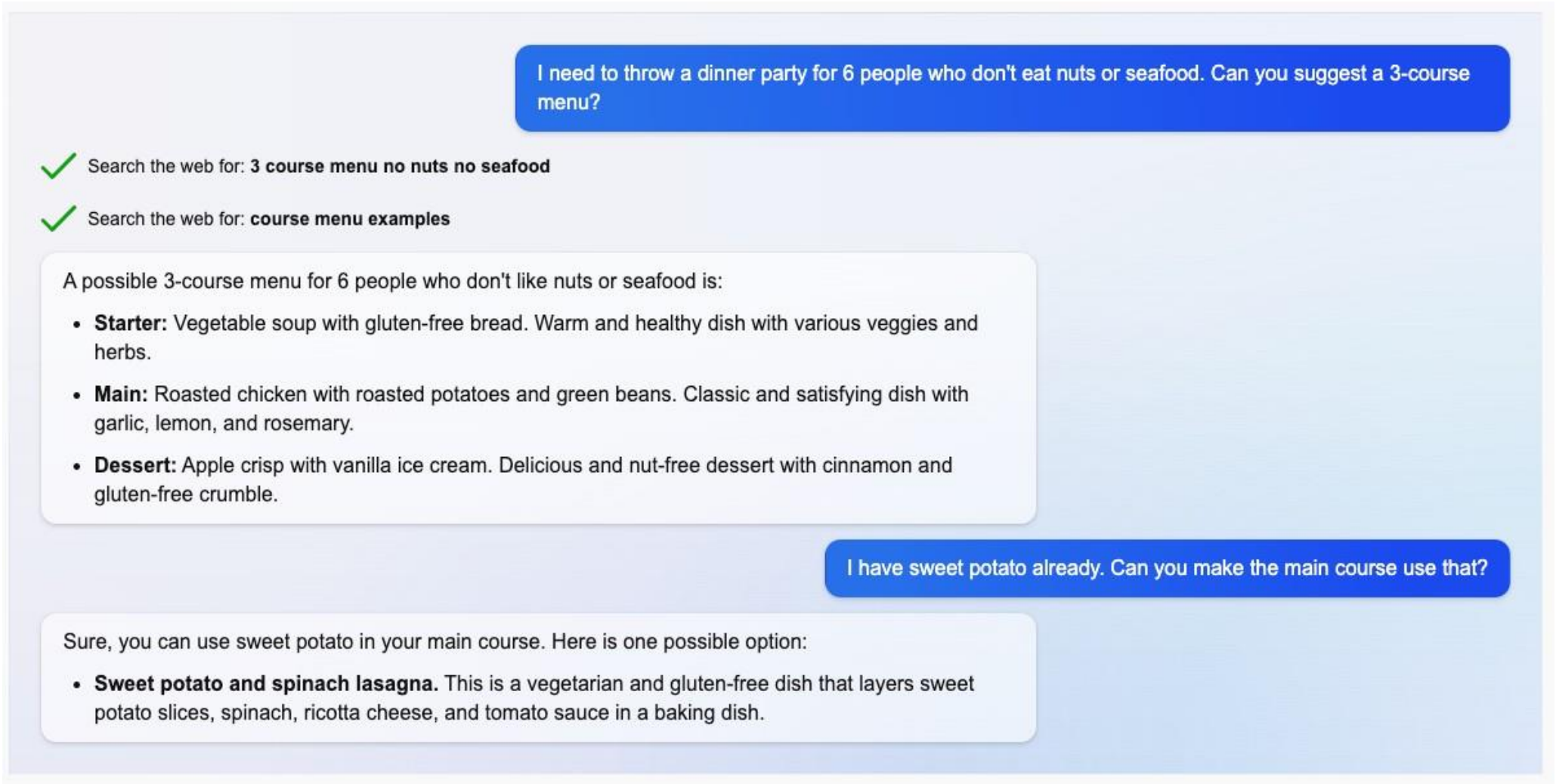
Lee, Mina, Percy Liang, and Qian Yang. "Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities." CHI 2022

Examples of HAI Collaboration: Co-pilot



https://github.com/features/copilot

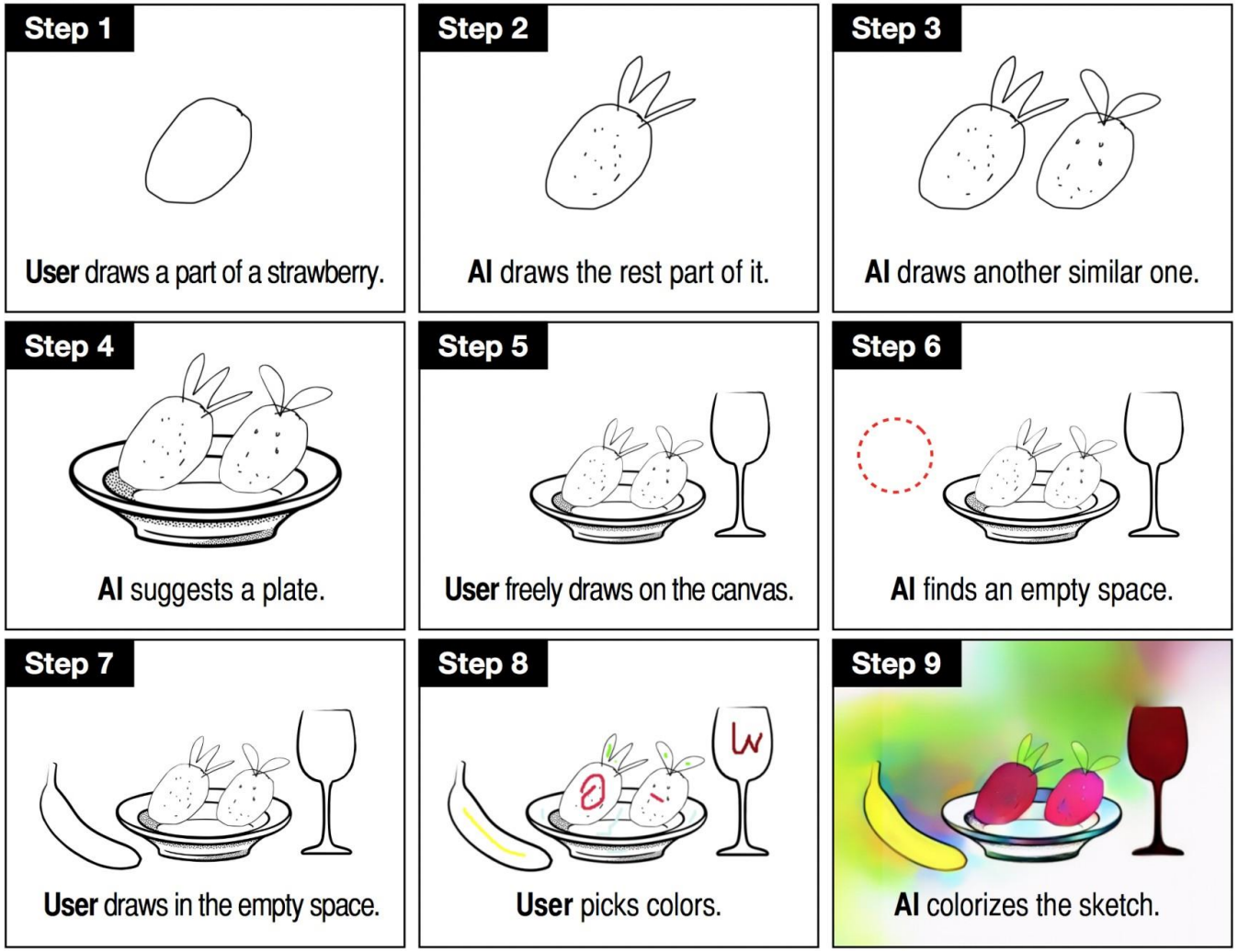
Examples of HAI Collaboration: Co-planning



Human-AI Collaboration: Key Aspects

- **Complementary performance**: Leverage the strengths of both AI and humans, to achieve better outcomes than either could accomplish alone.
- **Alignment**: AI should behave in a ways that humans expect their “teammate” to behave.
- **Cooperative interaction**: The interaction between human and AI should work in a way that’s both intuitive for the human, and compatible to model characteristics.
- **Reasonable task allocation**: Assign tasks that the model can perform.
- **Appropriate trust**: humans should follow AIs when they are correct, but identify and correct their mistakes otherwise.

Complementary Performance in Co-drawing

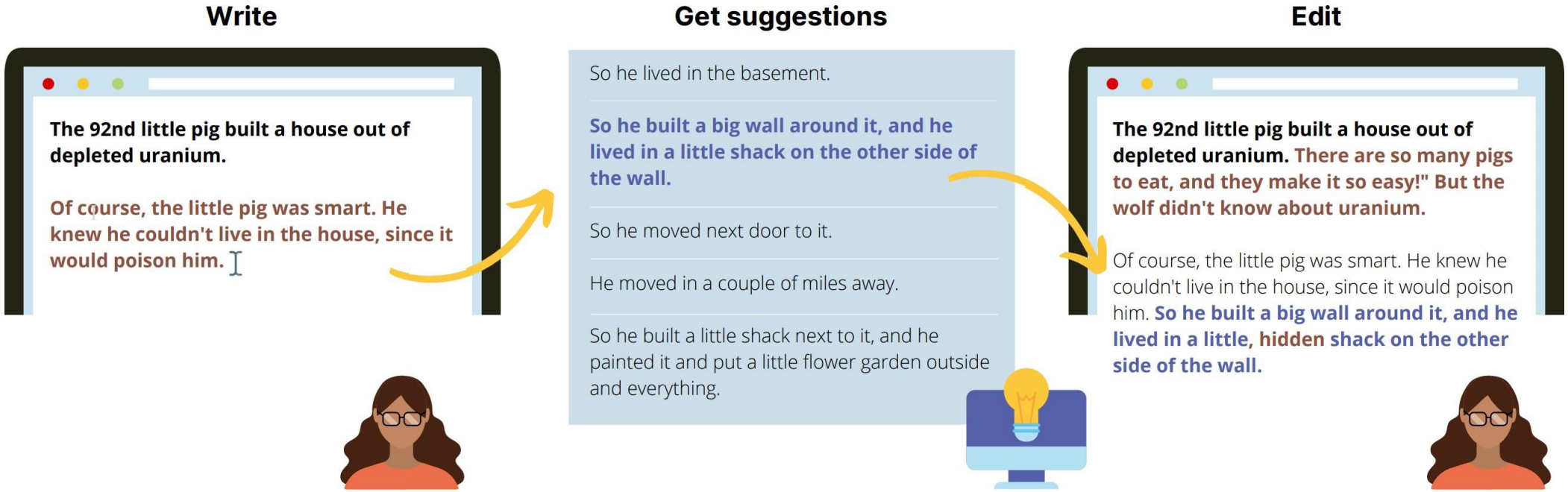


Human (*good at taking visual signals, draw rough shapes, annotate colors*)**:** lead the creation

Model (*repetition, detect space, detect objects & find similar objects*)**:** Automate repetitive tasks like draw the dots, duplicate the figure, fill in the color, suggest places to draw more

Oh, Changhoon, et al. "I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence." CHI 2018

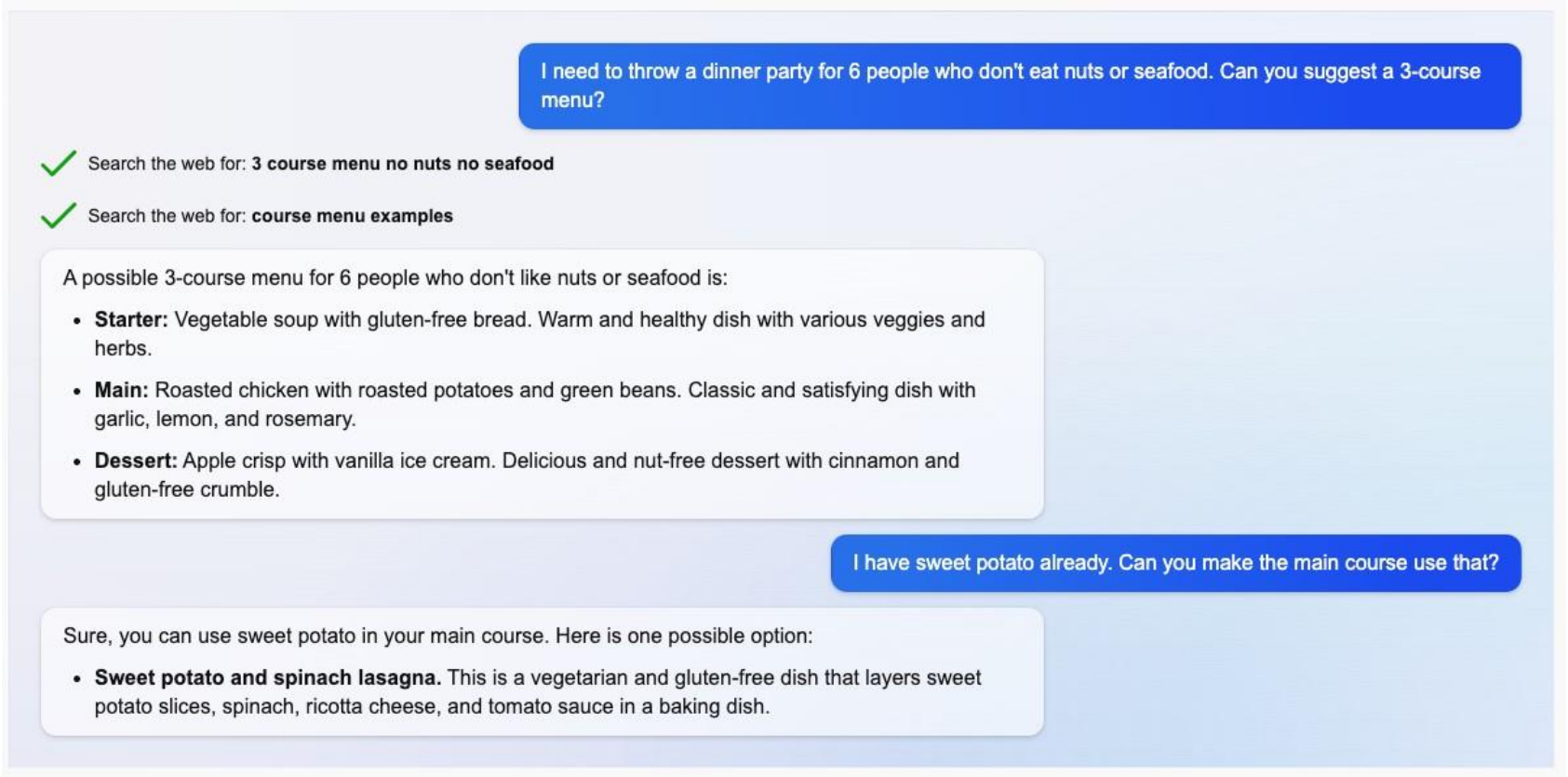
Complementary Performance in Co-authoring



Human *(good at logical reasoning and consistency in long doc, know what they want)*
Lead the writing, edit the model suggestions.
Model *(good at quick generate text many versions of text based on local context)*
Suggest next sentences, help write faster & overcome writer's block

Lee, Mina, Percy Liang, and Qian Yang. "Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities." CHI 2022

Complementary Performance: GPT



Human *(know what they need)* Iteratively refine their search query

Bing search chat *(have access to web)* Provide answer given the search constraint

https://github.com/features/copilot

Align Models with Humans through Human Feedback

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

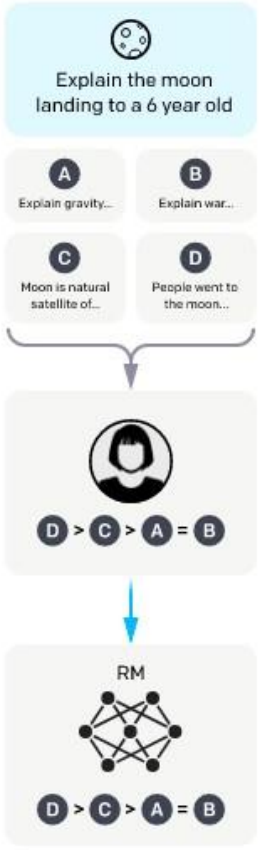
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

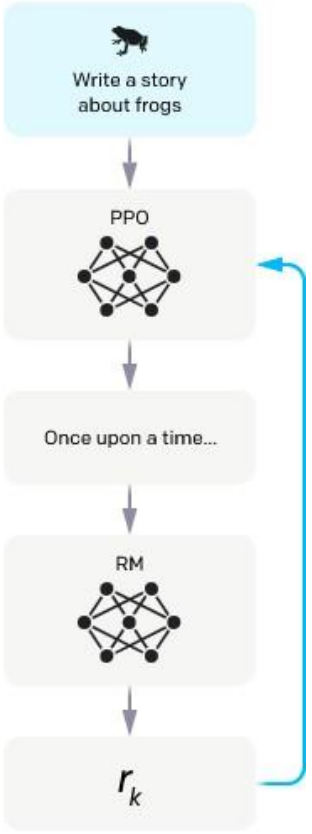
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Work with InstructGPT: Prompt Engineering

Prompts involve instructions and context passed to a LM to achieve a desired task **Prompt engineering** is the practice of developing and optimizing prompts to efficiently use language models (LMs) for a variety of applications

```
Prompt engineering is a process of creating a set of prompts, or
questions, that are used to guide the user toward a desired outcome. It
is an effective tool for designers to create user experiences that are
easy to use and intuitive. This method is
often used in interactive design and software development, as
it allows users to easily understand how to interact with a
system or product...
```

Elements of a Prompt

Prompts involve instructions and context passed to a LM to achieve a desired task

Prompt engineering is the practice of developing and optimizing prompts to efficiently use language models (LMs) for a variety of applications

A prompt is composed with the following components:

Instructions

Input context, & data

Output indicator, & output

```
Classify the text into neutral, negative or positive

Text: I think the food was great.
Sentiment: positive
```


Note the next-word-prediction structure!

Know what humans want through prompt cues

Sentiment analysis classifier

```
Decide whether a Tweet's sentiment is
positive, neutral, or negative.

Tweet: "I loved the new Batman movie!"
Sentiment: Positive
```

Q&A

Targeted world knowledge

```
Chatbot: I am a ML/AI language model tutor
You: What is a language model?
Chatbot: A language model is a statistical
model that describes the probability of a
word given the previous words.
```

Translation / NL2code

```
Create a SQL request to find all users who
live in California and have over 1000
credits: SELECT * FROM users WHERE
state='CA' AND credits > 1000;
```

Semantic of query

Syntax of code

Summarization

```
A neutron star is the collapsed core of a massive
supergiant star, which had a total mass of between 10
and 25 solar masses, possibly more if the star was
especially metal-rich.[1] Neutron stars are the
smallest and densest stellar objects, excluding black
holes and hypothetical white holes, quark stars, and
strange stars.[2] Neutron stars have a radius on the
order of 10 kilometres (6.2 mi) and a mass of about 1.4
solar masses.[3] They result from the supernova
explosion of a massive star, combined with
gravitational collapse, that compresses the core past
white dwarf star density to that of atomic nuclei.

TL;DR: A neutron star is the collapsed core of a
massive supergiant star. These ultra-dense objects are
incredibly fascinating due to their strange properties
and their potential for phenomena such as extreme
gravitational forces and a strong magnetic field.
```

The semantic of TL;DR

GPT-3 API, from OpenAI: https://beta.openai.com/examples

Human-AI Collaboration: Key Aspects

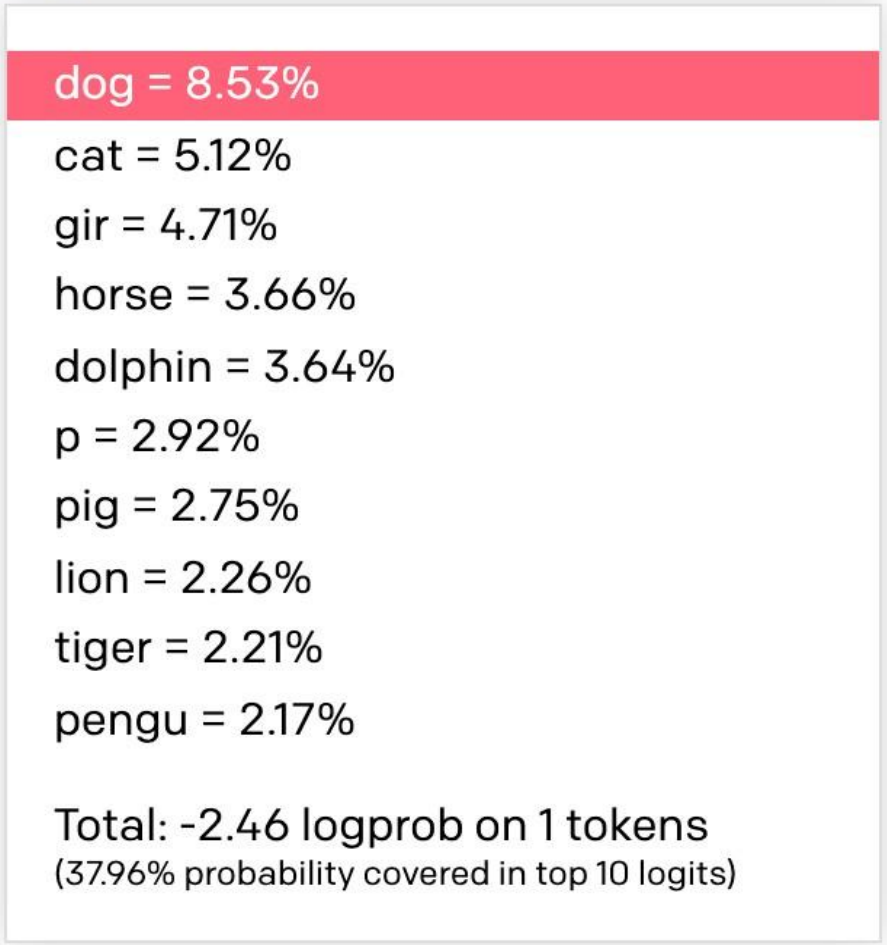
- **Complementary performance**: Leverage the strengths of both AI and humans, to achieve better outcomes than either could accomplish alone.
- **Alignment**: AI should behave in a ways that humans expect their “teammate” to behave.
- **Cooperative interaction**: The interaction between human and AI should work in a way that’s both intuitive for the human, and compatible to model characteristics.
- **Reasonable task allocation**: Assign tasks that the model can perform.
- **Appropriate trust**: humans should follow AIs when they are correct, but identify and correct their mistakes otherwise.

Work with LLM - Account for Model Characteristics

- Language models are not deterministic.
- Language models are lack of reasoning capabilities.
- Language models sometimes cannot understand instructions.

Language model is not deterministic.

My favorite animal is a dog



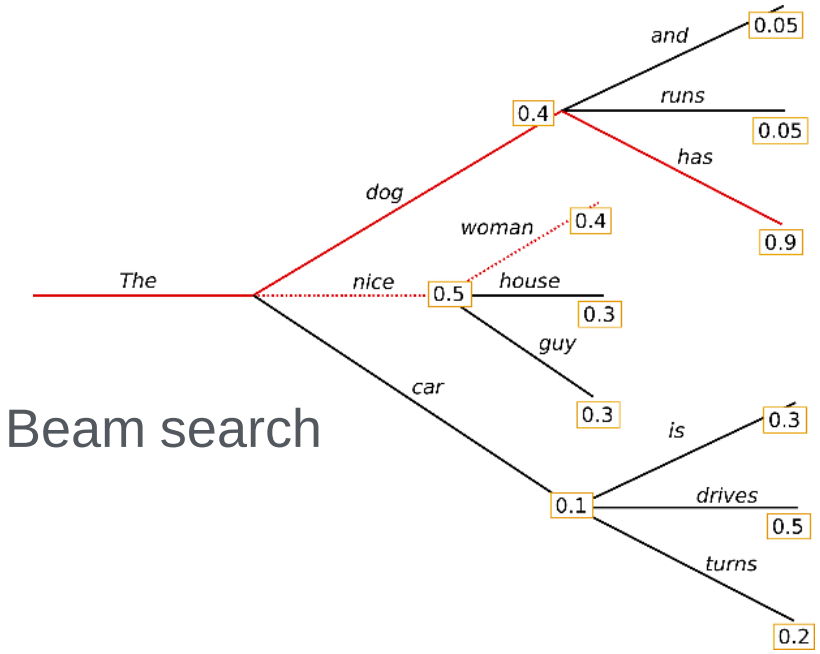
How should we deal with such “randomness”?

Depends on the task —remove, express, or exploit!

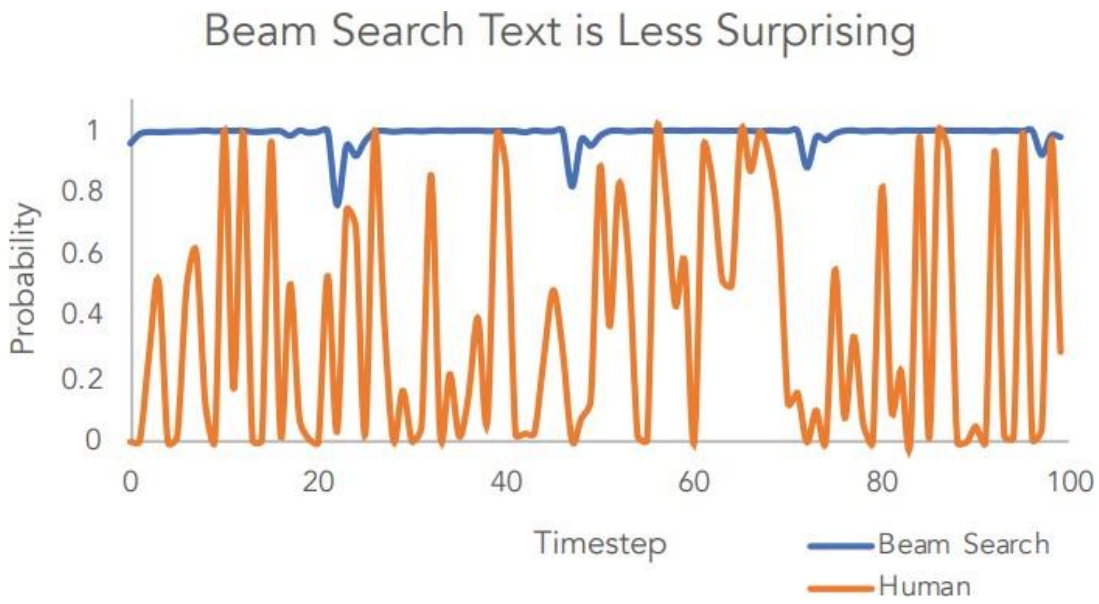
Non-deterministic LMs ➡️ Remove Uncertainty

When tasks need certainty,
(e.g. write formal documents, need to maximize grammatical correctness, do classification)
remove uncertainty through:

parameters (temperature=0)**, less surprising sampling** (e.g. beam search or look ahead)



Beam search

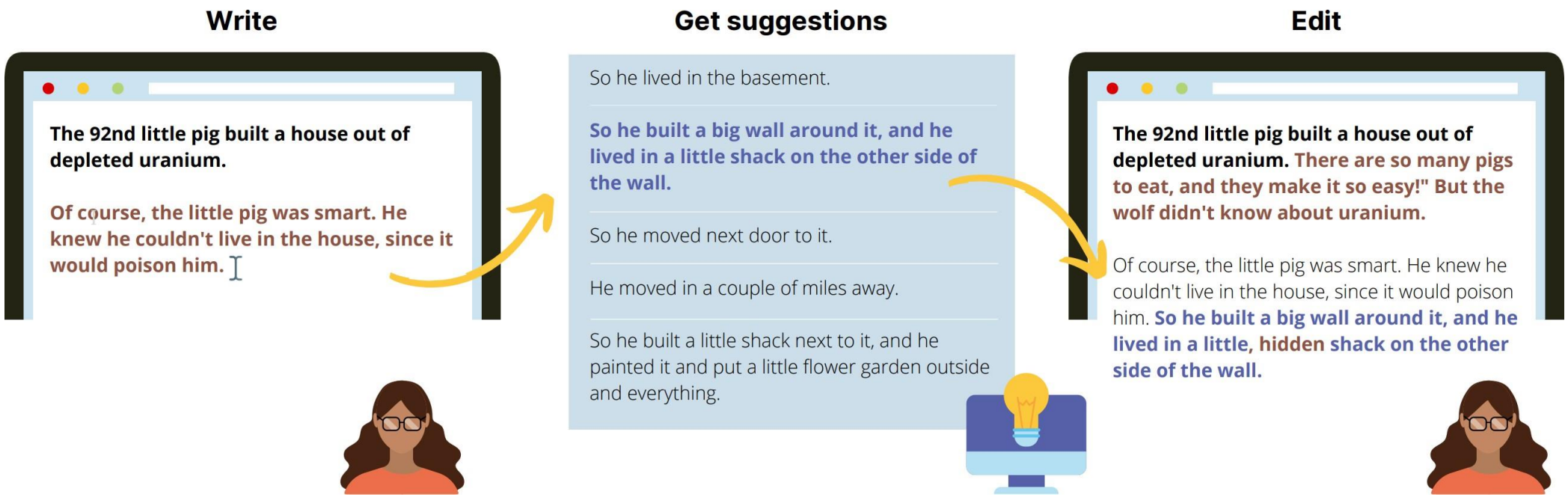


https://huggingface.co/blog/how-to-generate
Holtzman, Ari, et al. "The curious case of neural text degeneration." *arXiv preprint arXiv:1904.09751* (2019).

Non-deterministic LMs ➡️ Exploit Uncertainty

When tasks need creativity,
(e.g. creative writing, ideation, etc.) exploit uncertainty through:

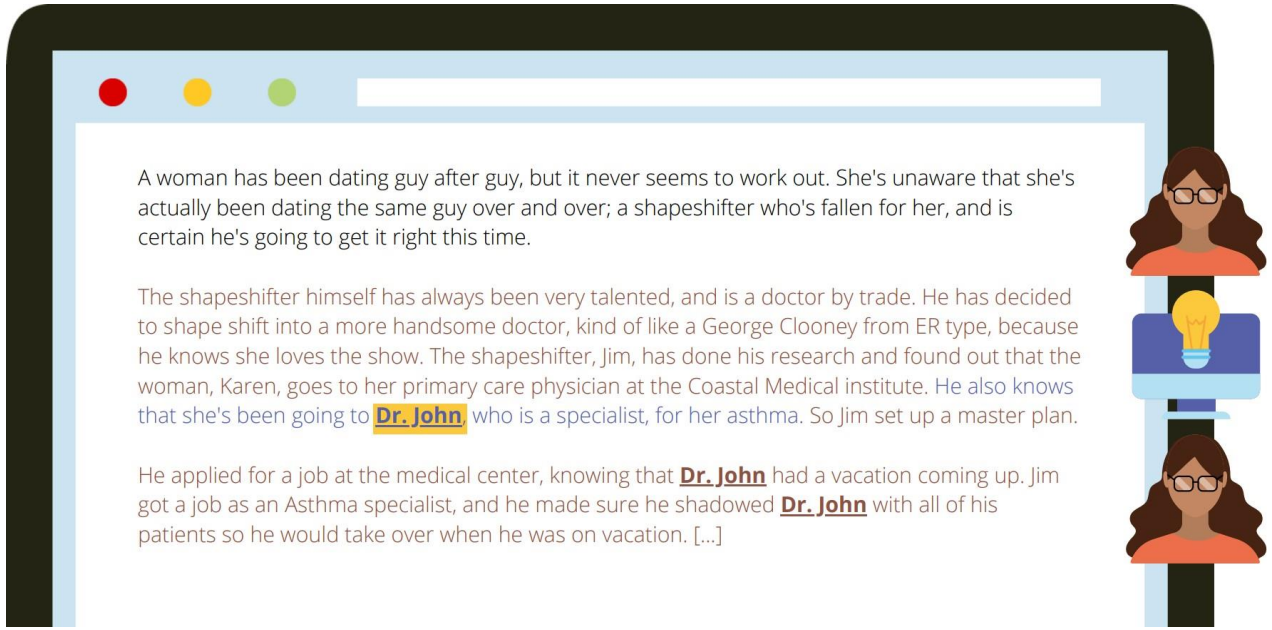
parameters (temperature=1)**, re-running the generation multiple times**



Lee, Mina, Percy Liang, and Qian Yang. "CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities." *CHI 2022* .

Non-deterministic LMs ➡️ Exploit Uncertainty

When tasks need creativity,
(e.g. creative writing, ideation, etc.)
exploit uncertainty through:
parameters (temperature=1)**, re-running the generation multiple times**



“Model hallucination is not a bug but a feature”: Can be **inspirations** to humans.

Lee, Mina, Percy Liang, and Qian Yang. "CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities." *CHI 2022* .

LLMs: No reasoning capability

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

LLMs are designed to grasp the form of language, rather than the meaning. They can struggle on tasks like sequential arithmetic problems, multi-hop question answering, recognizing and comparing sentences, or those that require branching logic.

Holtzman, Ari, et al. "The curious case of neural text degeneration." *arXiv preprint arXiv:1904.09751* (2019).

Lack-of-reasoning ➡️ Chain-of-Thought

Plain prompting
Model cannot do reasoning based on word distribution

```
I went to the market and bought 10 apples. I gave 2 apples to
the neighbor and 2 to the repairman. I then went and bought 5
more apples and ate 1. How many apples did I remain with?
11 apples
```

CoT prompting
With “step-by-step”, can make the final output depend on intermediate ones

```
I went to the market and bought 10 apples. I gave 2 apples to
the neighbor and 2 to the repairman. I then went and bought 5
more apples and ate 1. How many apples did I remain with?

Let's think step by step.
First, you started with 10 apples. You gave away 2 apples to
the neighbor and 2 to the repairman, so you had 6 apples left.
Then you bought 5 more apples, so now you had 11 apples.
Finally, you ate 1 apple, so you would remain with 10 apples.
```

Prompting can be further improved by ***instructing the model to reason*** about the task.

Cannot understand instruction ➡️ Few-shot

Zero-shot

Natural language descriptions only

Few-shot

Description + a few example (3-100)

[5-10 is most common]

```
1  Find the nationality of people:
2  Marie Curie =>
```

Task instruction

Task

```
1  Find the nationality of people:
2  Albert Einstein => German
3  Alan Turing => English
4  Mahatma Gandhi => Indian
5  Marie Curie =>
```

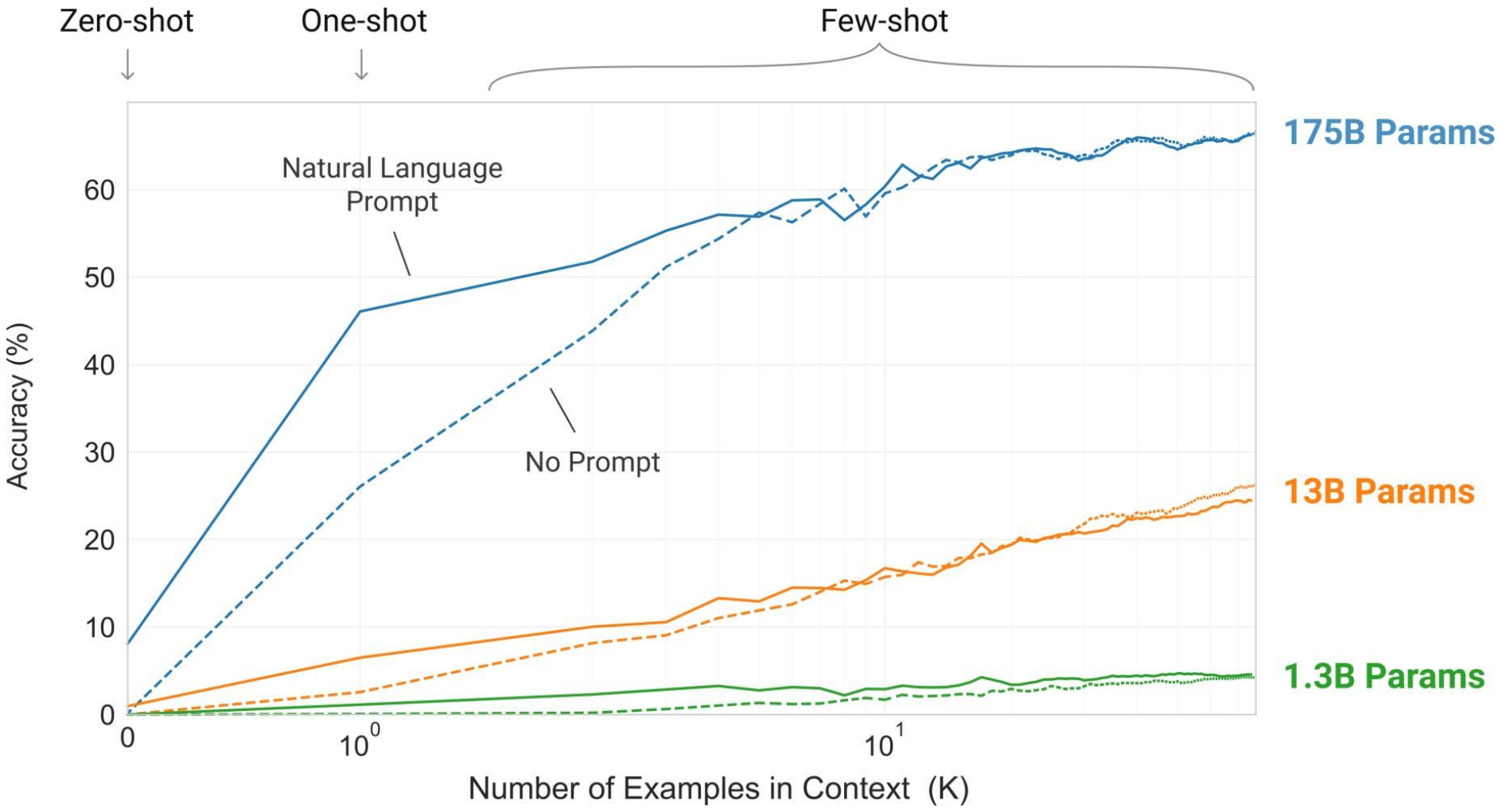
Task instruction

Examples

Task

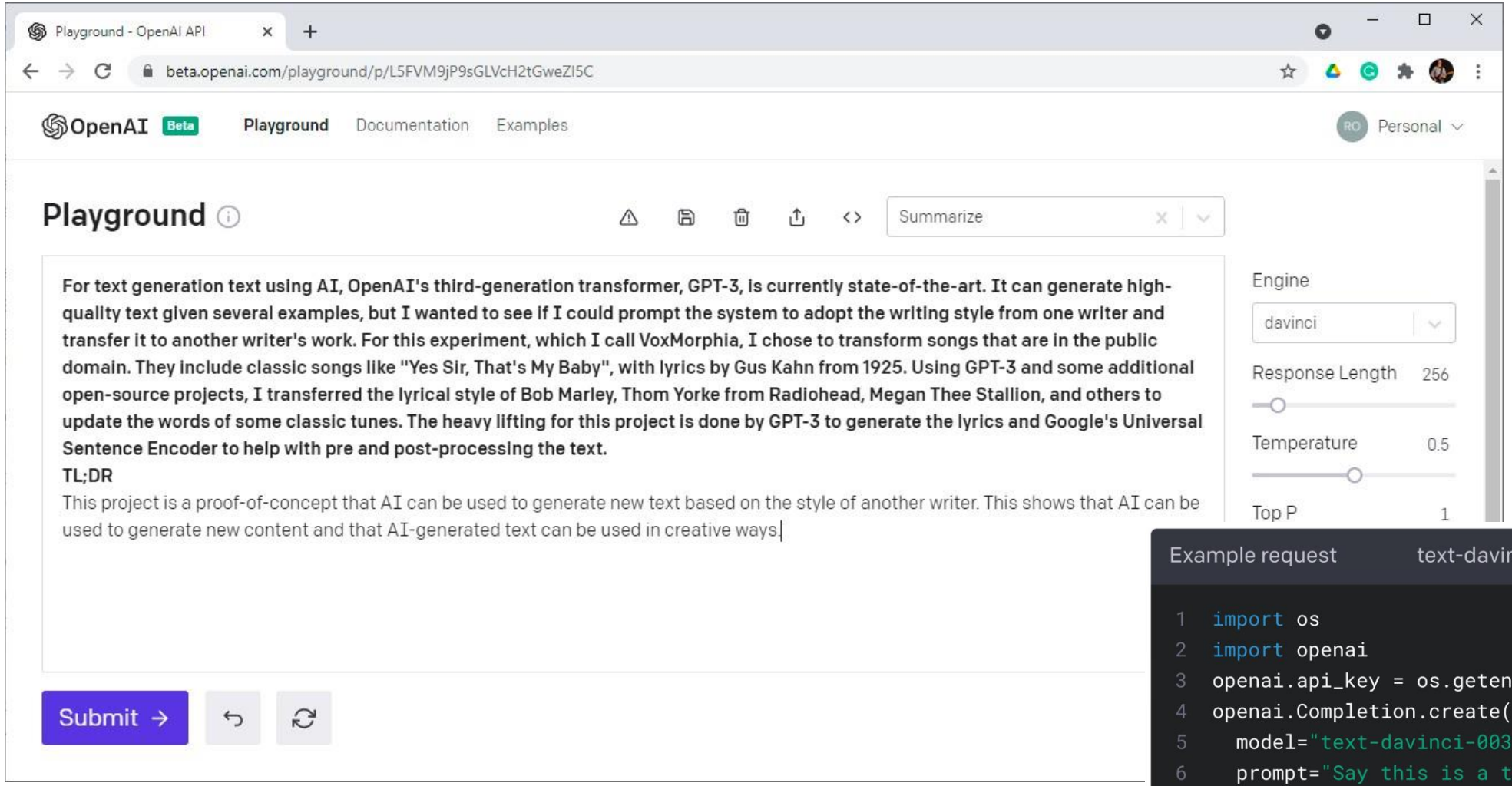
Brown, Tom, et al. “Language Models are Few-Shot Learners.” *NeurIPS 2020.*

Gain in few-shot: You only need some examples



Brown, Tom, et al. "Language Models are Few-Shot Learners." *NeurIPS 2020.*

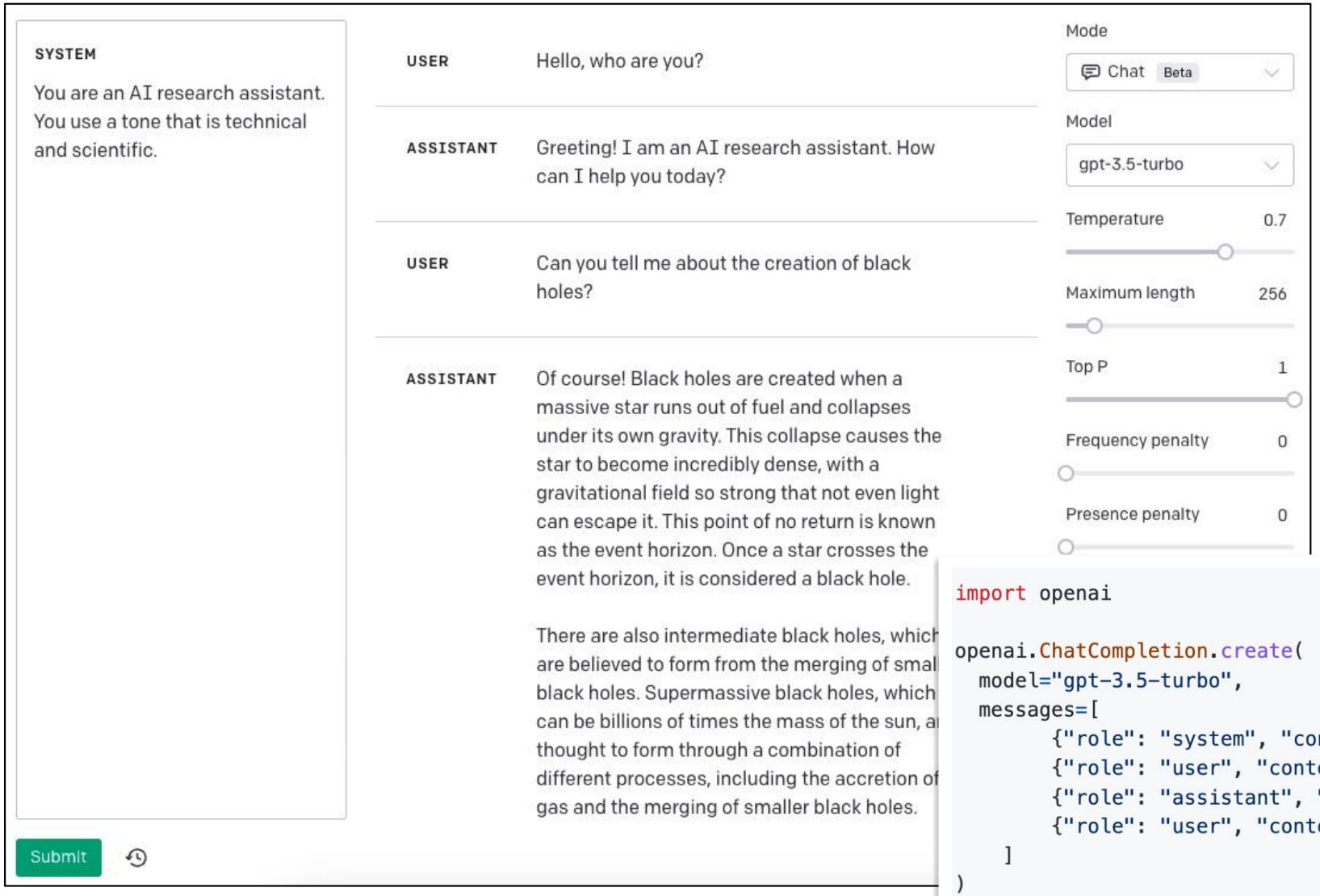
ChatGPT vs. InstructGPT: Different Interfaces



Example request text-davinci-003 python Copy

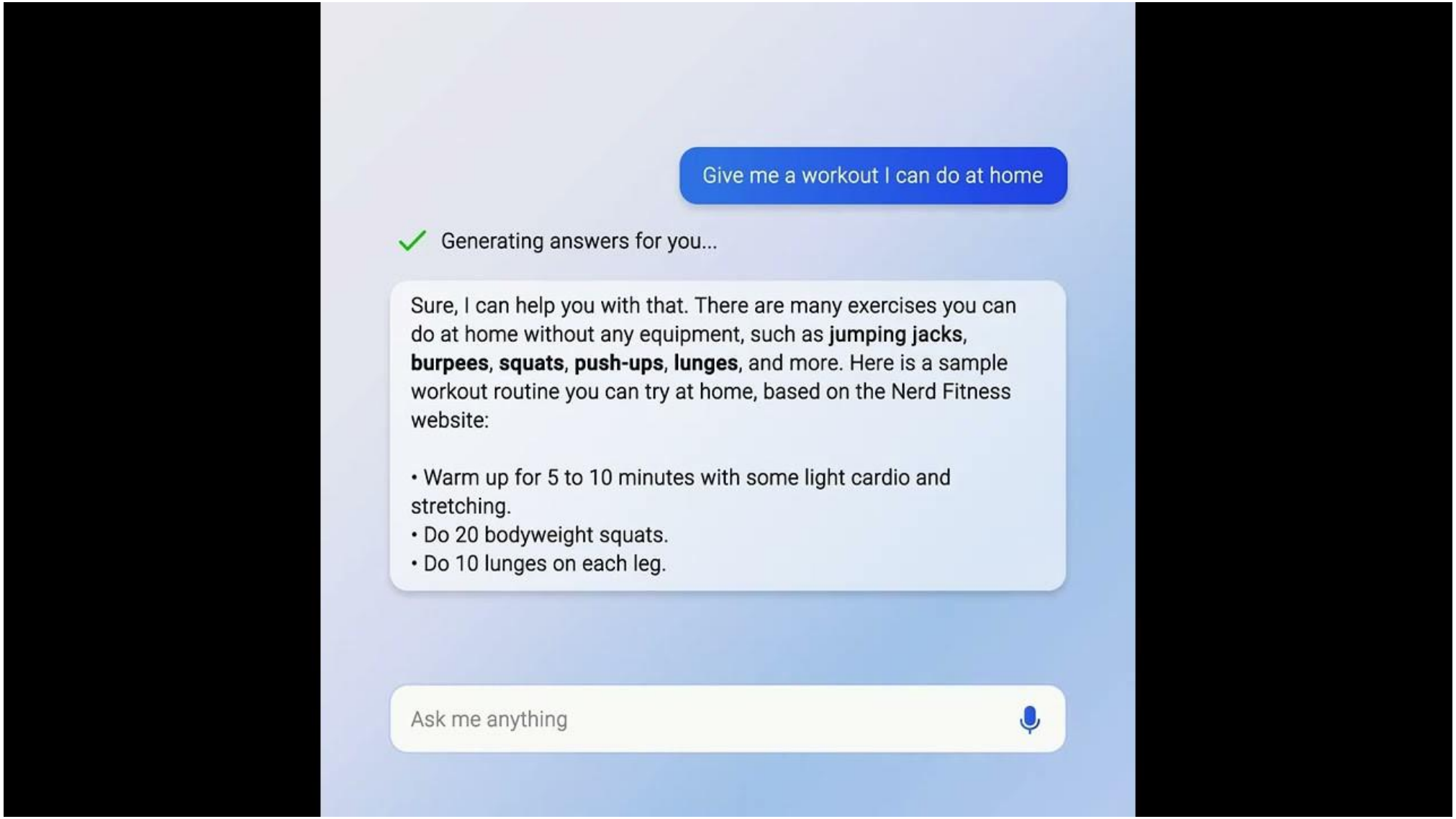
```python
import os
import openai
openai.api_key = os.getenv("OPENAI_API_KEY")
openai.Completion.create(
  model="text-davinci-003",
  prompt="Say this is a test",
  max_tokens=7,
  temperature=0
)
```

ChatGPT vs. InstructGPT: Different Interface

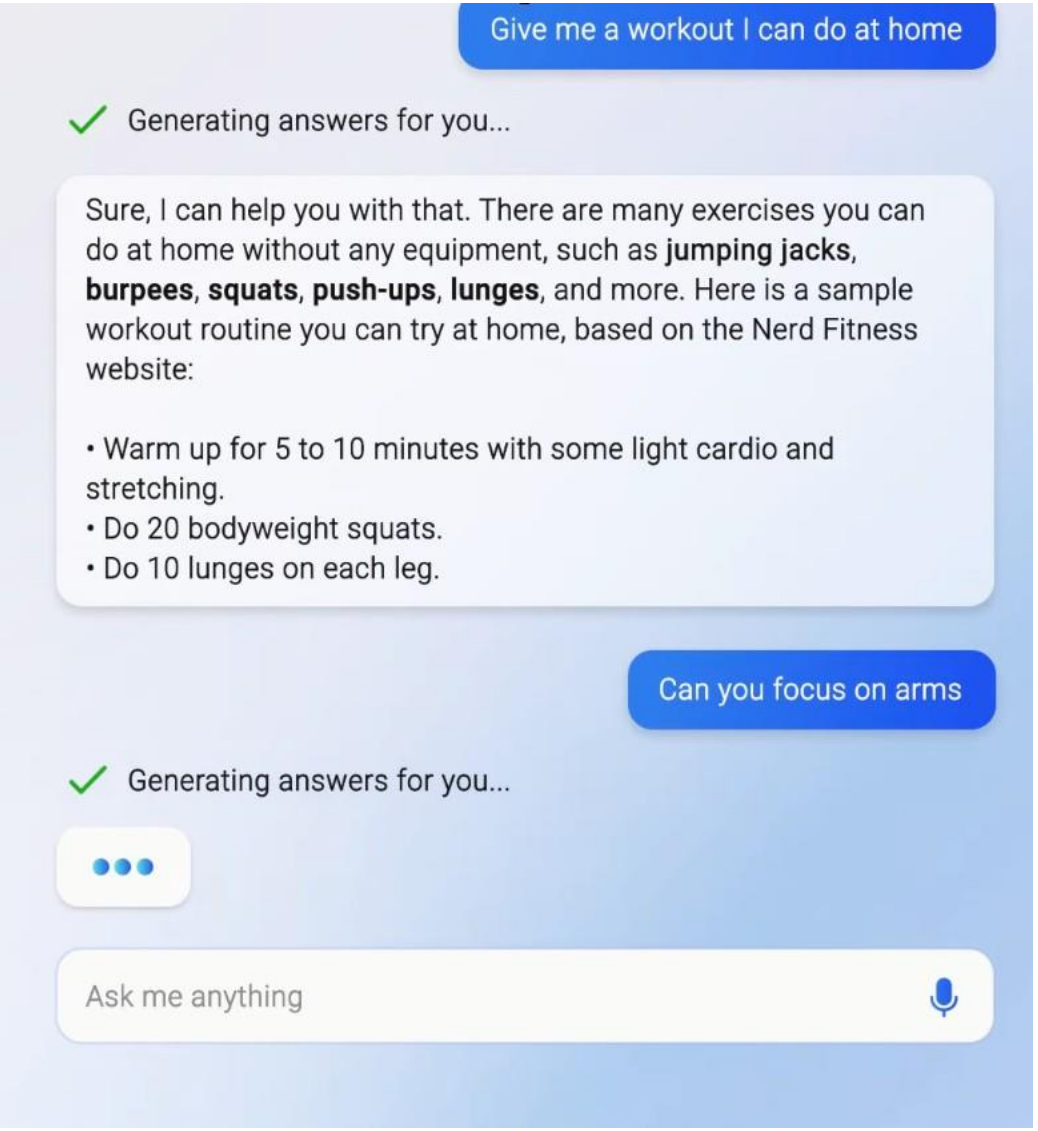



```
import openai

openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
        {"role": "system", "content": "You are an AI research assistant. You use a tone th
        {"role": "user", "content": "Hello, who are you?"},
        {"role": "assistant", "content": "Greeting! I am an AI research assistant. How car
        {"role": "user", "content": "Can you tell me about the creation of black holes?"}
    ]
)
```

Interface changes how you use the AI!



Interface changes how you use the AI!



e.g., Compared to traditional search, Bing w/ ChatGPT is more **conversational** and **interactive**…

You use more **complete sentences**.

You get **a sense of talking** to someone and therefore become more polite (“can you…”)

You can follow up more naturally, in a **context-aware way**.

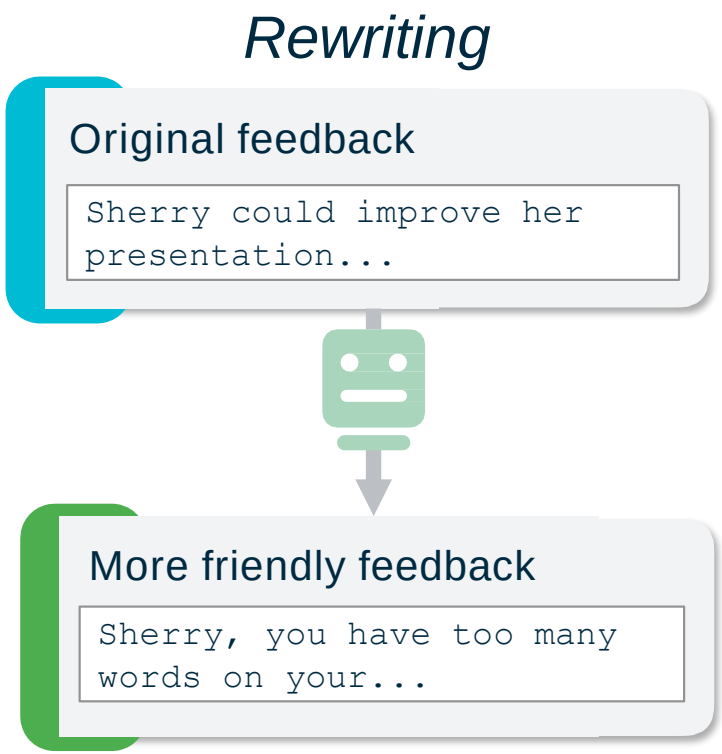
You use more **“engaging”** prompts, e.g. ask questions, provide commands, do reflections, etc.

Human-AI Collaboration: Key Aspects

- **Complementary performance**: Leverage the strengths of both AI and humans, to achieve better outcomes than either could accomplish alone.
- **Alignment**: AI should behave in a ways that humans expect their “teammate” to behave.
- **Cooperative interaction**: The interaction between human and AI should work in a way that’s both intuitive for the human, and compatible to model characteristics.
- **Reasonable task allocation**: Assign tasks that the model can perform.
- **Appropriate trust**: humans should follow AIs when they are correct, but identify and correct their mistakes otherwise.

A case study: LLMs as multiple teammates.

Single models struggle with complex tasks. Example: Peer review rewriting

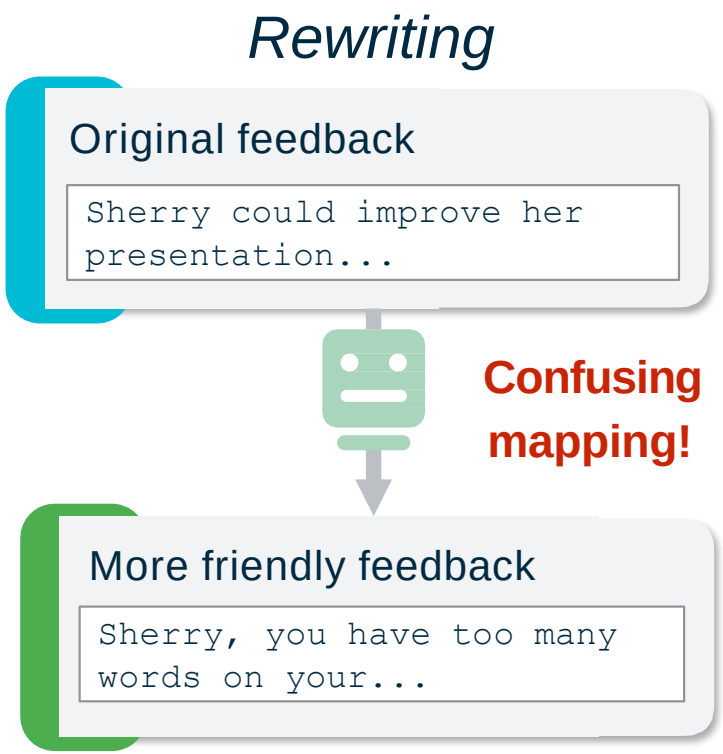



```
Given the following feedback, rewrite it into a friendly
paragraph with concrete suggestions for each of Alex's
presentation problems.

Original feedback:
Sherry could improve her presentation skills. She has too
much text on her slides. Her presentation meanders from
topic to topic without a clear structure. She also does
not engage with her audience when she presents.

More friendly feedback:
Sherry, you have too many words on your slides. You should
use images and bullet points to help get your message
across. You should have a clear structure for the
presentation. You should also engage with your audience.
```

Wu, Tongshuang, Michael Terry, and Carrie Jun Cai. "Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts." *CHI Conference on Human Factors in Computing Systems*. 2022.

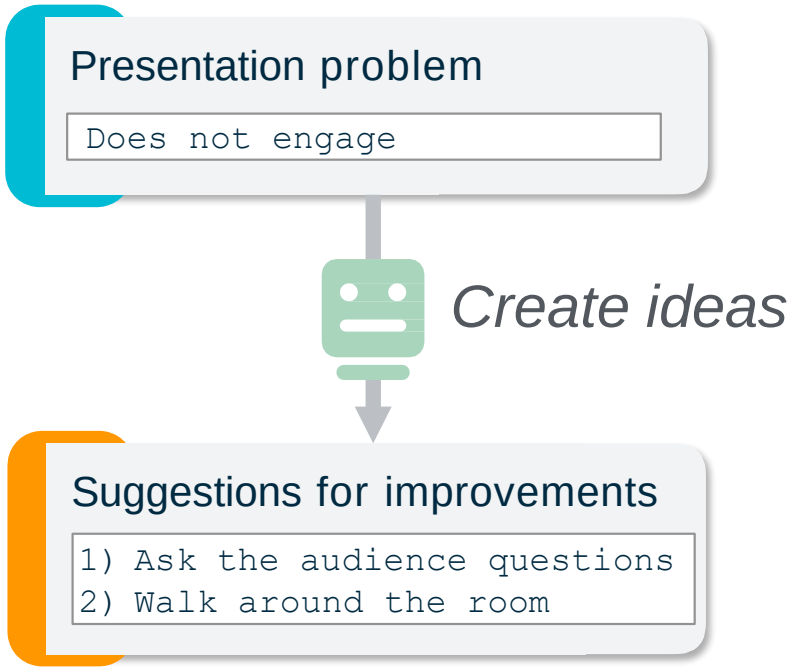
The input-output mapping is convoluted.

Example: Peer review rewriting



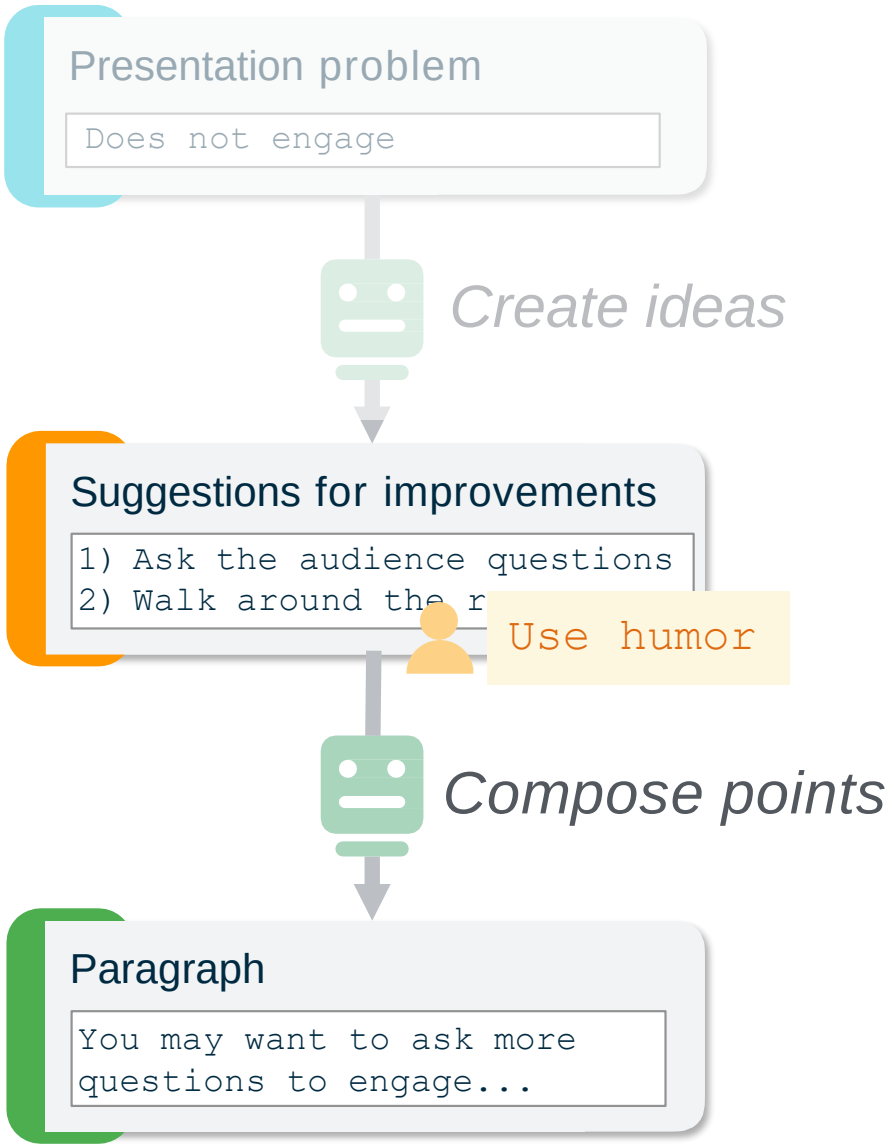

```
Given the following feedback, rewrite it into a friendly
paragraph with concrete suggestions for each of Alex's
presentation problems.

Original feedback:
Sherry could improve her presentation skills. She has too
much text on her slides. Her presentation meanders from
topic to topic without a clear structure. She also does
not engage with her audience when she presents.

More friendly feedback:
Sherry, you have too many words on your slides. You should
use images and bullet points to help get your message
across. You should have a clear structure for the
presentation. You should also engage with your audience.
```

Small tasks are more interpretable and controllable.




```
Given the Presentation problem, the following
is a list of improvement suggestions.

Problem: Does not engage
Suggestions for improvements:
1) Ask the audience questions
2) Use humor
```

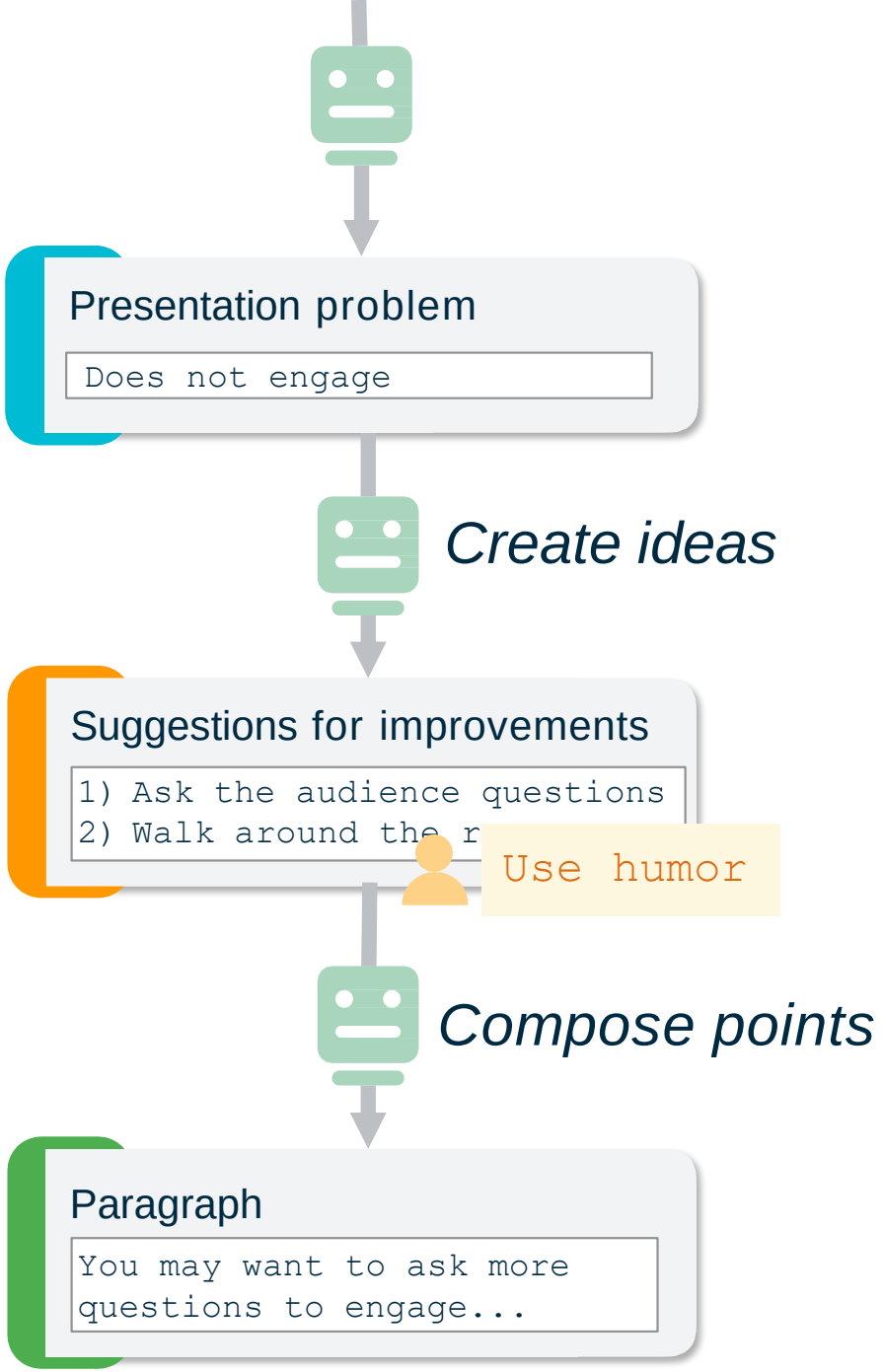
The fix can be propagated to related sub-tasks!




```
Write one friendly paragraph to cover all the suggestions.

Suggestions:
1) Ask the audience questions
2) Use humor

Paragraph
```

```
You may want to ask more questions to engage the audience. Humor always helps!
```

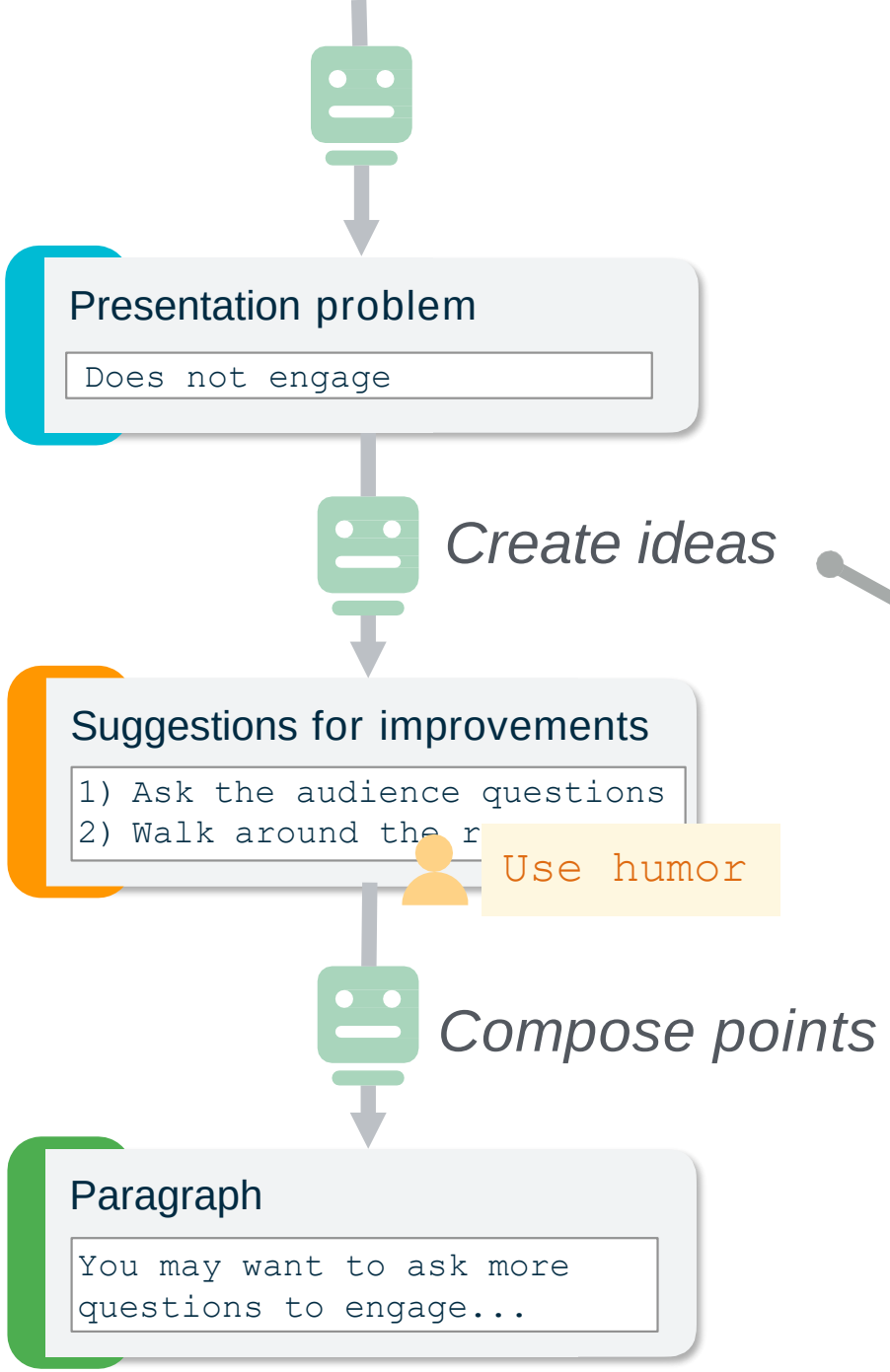
Brown, Tom B., et al. "Language models are few-shot learners." arXiv 2020



Chain =

Independent LLM runs per sub-task

+ inter-step transformation



Chain =

Independent LLM runs per sub-task

1. Identify all presentation problems
2. Ideate suggestions per problem
3. Compose them into a paragraph

+ inter-step transformation

A three-step LLM Chain for peer review rewriting

Original feedback

```
Sherry could improve his presentation skills. He has too much text on his slides. His presentation
meanders from topic to topic without a clear structure. He also does not engage with his audience when he
presents.
```

1 Identify all presentation problems

2 Ideate suggestions per problem

3 Compose them into a paragraph

Friendly paragraph

A three-step LLM Chain for peer review rewriting

Original feedback

Sherry could improve his presentation skills. He has too much text on his slides. His presentation meanders from topic to topic without a clear structure. He also does not engage with his audience when he presents.

Sherry's problems

- Too much text on slides
- No clear structure
- Does not engage with audience

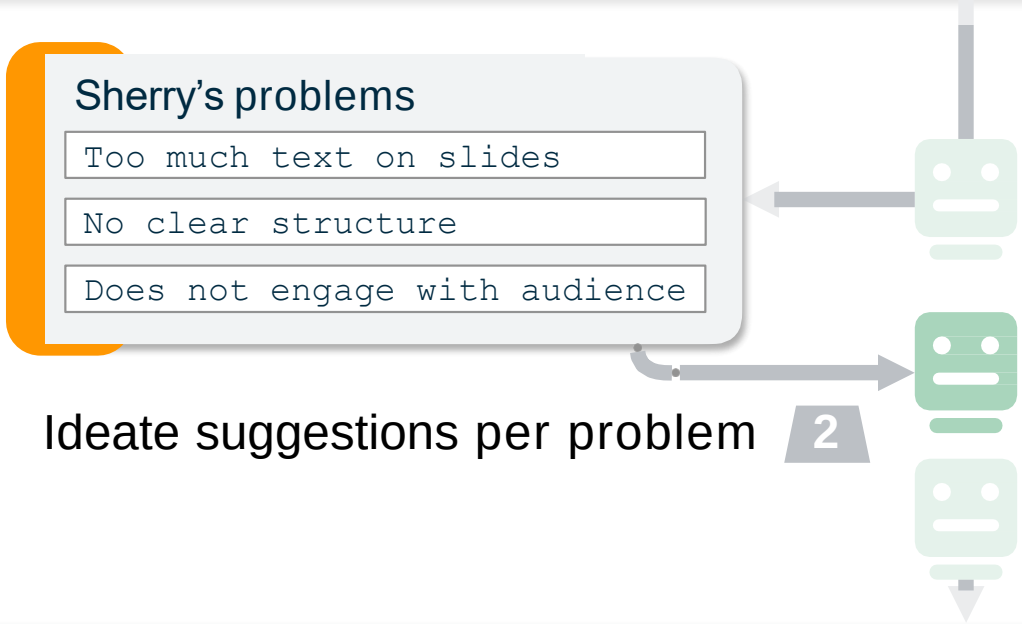
1 Identify all presentation problems

Friendly paragraph

A three-step LLM Chain for peer review rewriting

Original feedback

Sherry could improve his presentation skills. He has too much text on his slides. His presentation meanders from topic to topic without a clear structure. He also does not engage with his audience when he presents.

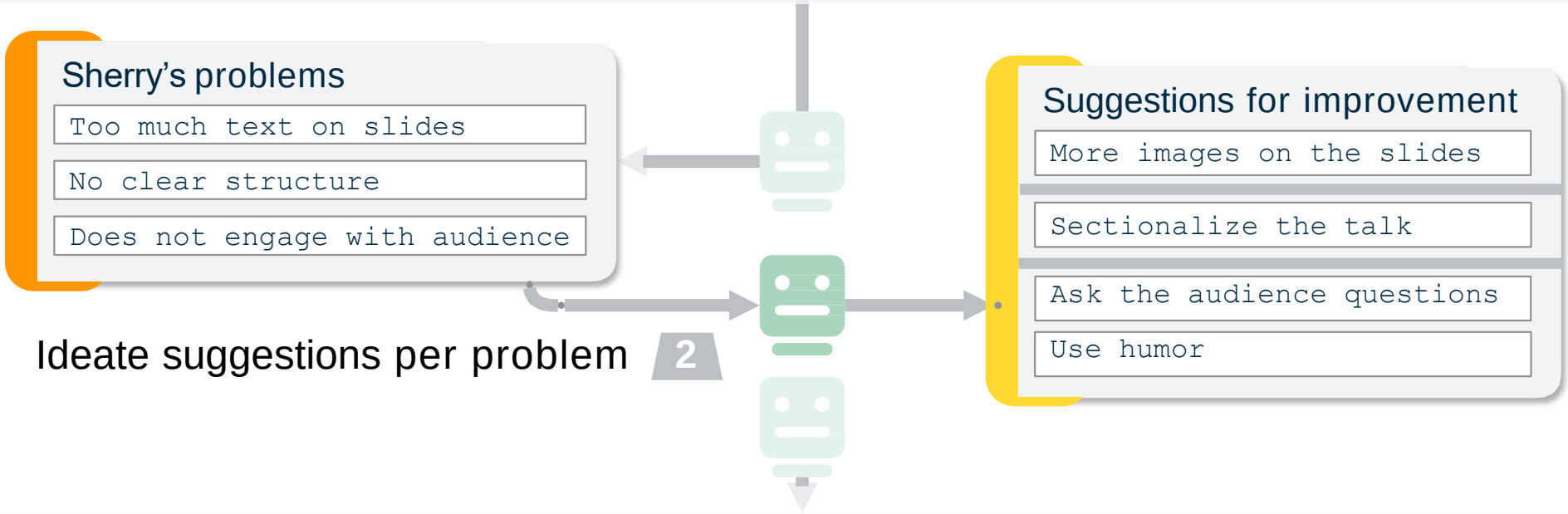


Friendly paragraph

A three-step LLM Chain for peer review rewriting

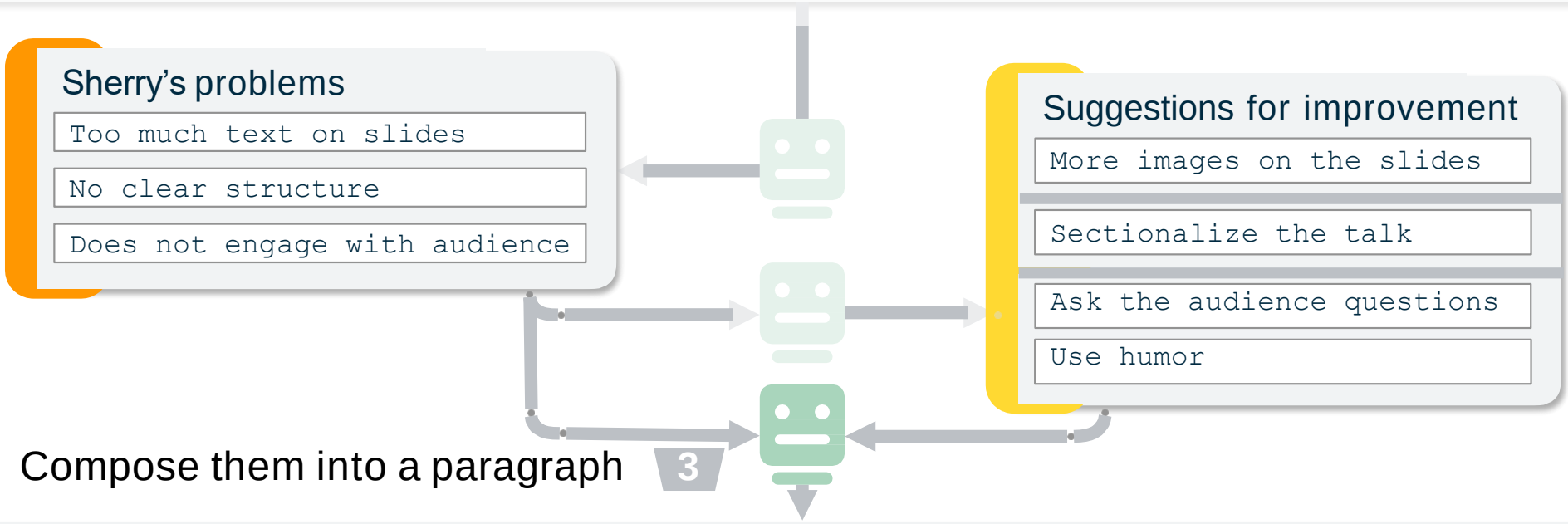
Original feedback

Sherry could improve his presentation skills. He has too much text on his slides. His presentation meanders from topic to topic without a clear structure. He also does not engage with his audience when he presents.



Friendly paragraph

A three-step LLM Chain for peer review rewriting

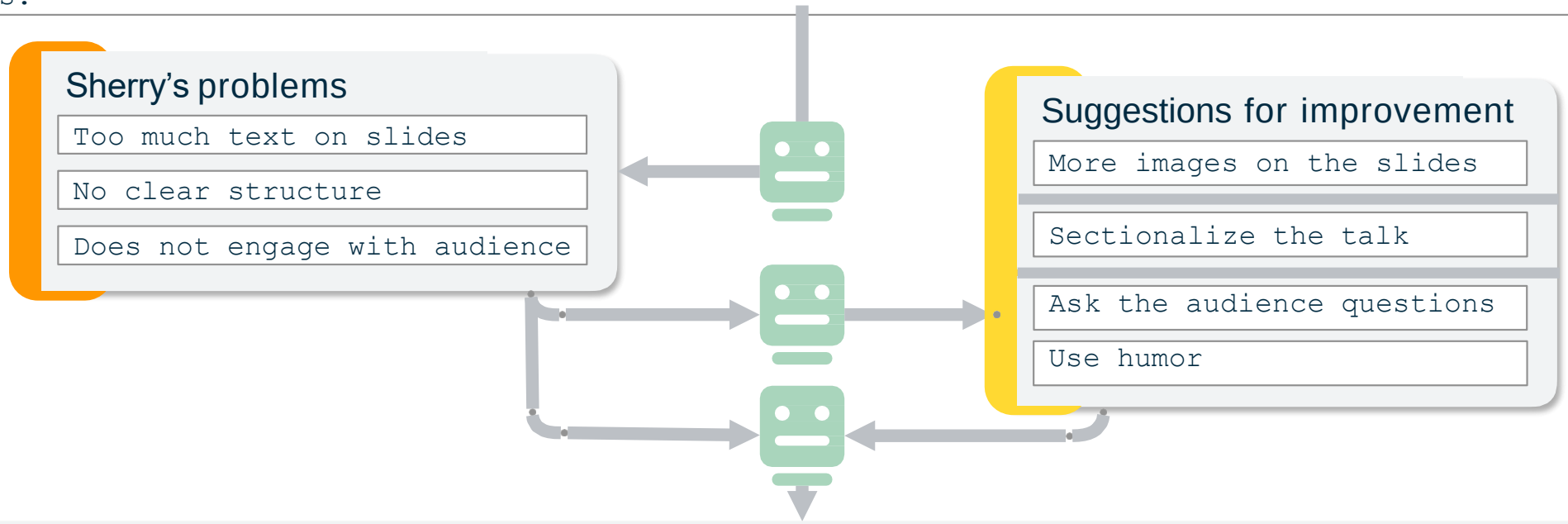


Original feedback

A three-step LLM Chain for peer review rewriting

Original feedback

Sherry could improve his presentation skills. He has too much text on his slides. His presentation meanders from topic to topic without a clear structure. He also does not engage with his audience when he presents.



Friendly paragraph

Sherry, your presentation was interesting! However, I noticed that you have a lot of information on your slides. It might be helpful to vary pictures with text so that it is easier to follow. Also, you might consider the flow of your theme. If it were me, I would have divided it into three sections and then used your conclusion. You may also want to add some humor, and ask more questions to engage the audience.

Chaining in Opensourcing

🦜🔗 LangChain

```
from langchain.prompts import PromptTemplate
from langchain.llms import OpenAI

llm = OpenAI(temperature=0.9)
prompt = PromptTemplate(
    input_variables=["product"],
    template="What is a good name for a company that makes {product}?",
)
```

```
second_prompt = PromptTemplate(
    input_variables=["company_name"],
    template="Write a catchphrase for the following company: {company_name}",
)
chain_two = LLMChain(llm=llm, prompt=second_prompt)
```

```
from langchain.chains import SimpleSequentialChain
overall_chain = SimpleSequentialChain(chains=[chain, chain_two], verbose=True)

# Run the chain specifying only the input variable for the first chain.
catchphrase = overall_chain.run("colorful socks")
print(catchphrase)
```


Read more on LangChain documentation

Chaining in Opensourcing

🦜🔗 LangChain

```
llm = OpenAI(temperature=0.9)
prompt = PromptTemplate(
    input_variables=["product"],
    template="What is a good name for a company that makes {product}?",
)
```

```
second_prompt = PromptTemplate(
    input_variables=["company_name"],
    template="Write a catchphrase for the following company: {company_name}",
)
chain_two = LLMChain(llm=llm, prompt=second_prompt)
```



```
> Entering new SimpleSequentialChain chain...

Cheerful Toes.

"Spread smiles from your toes!"

> Finished SimpleSequentialChain chain.

"Spread smiles from your toes!"
```

Read more on LangChain documentation

Chaining reminds us of workflows in crowdsourcing…

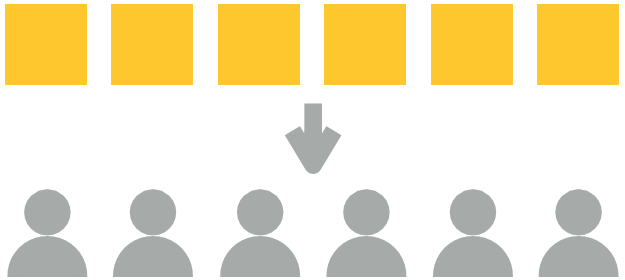
Crowdsourcing: Small Tasks, Many People

Combine many small tasks completed by independent workers.

e.g., text shortening

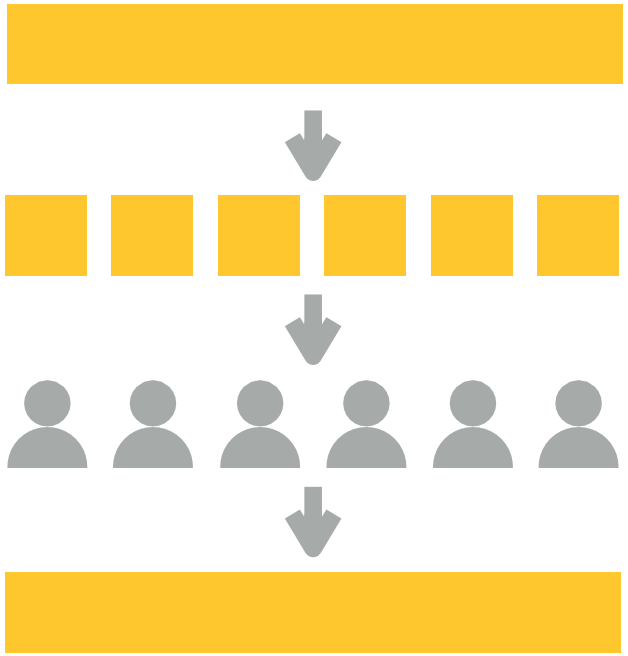
e.g., image labeling

e.g., translation



Crowdsourcing Workflows

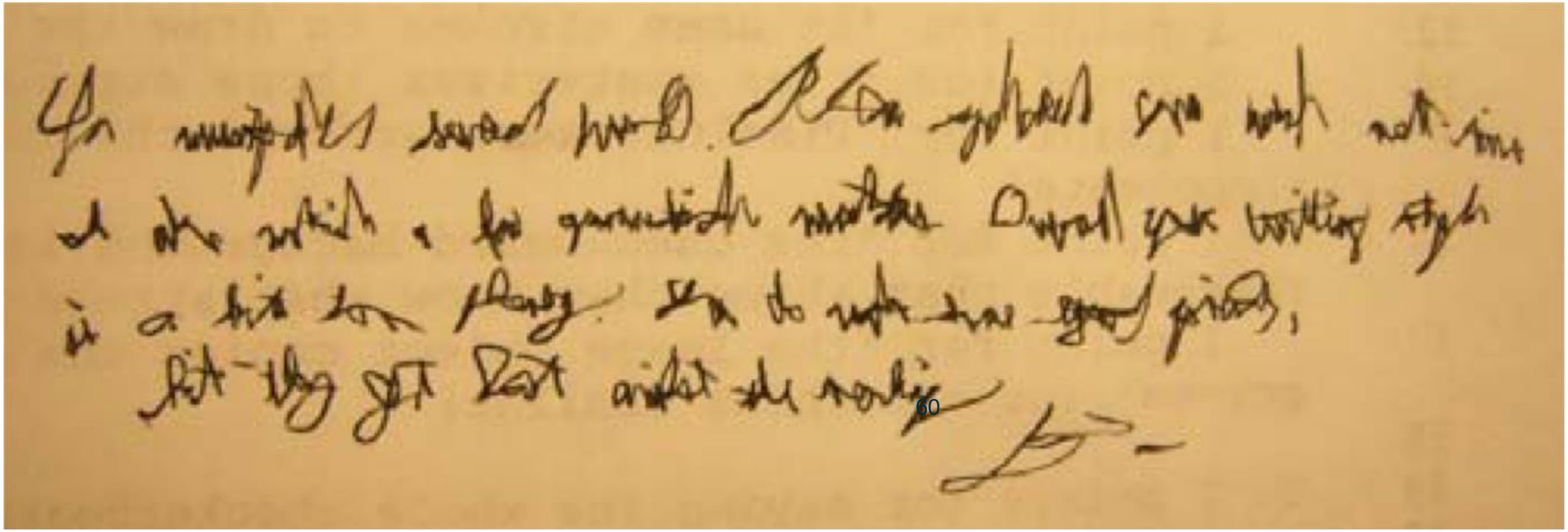
Pre-specified and computationally sequenced set of **decomposed** tasks that are assigned to **distributed** workers and combined to **reach a final goal.**



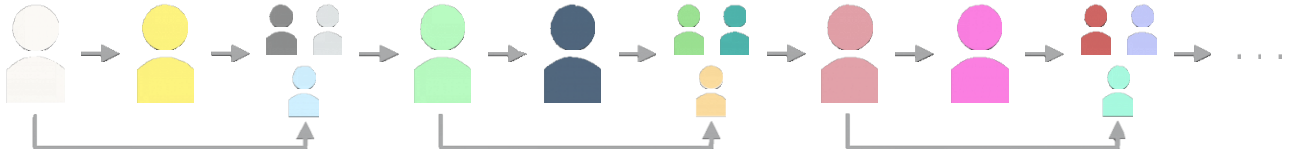
Crowdsourcing workflows have been used for:

- document editing
- email management
- text translation
- software development

Crowdsourcing workflow examples: Iterative

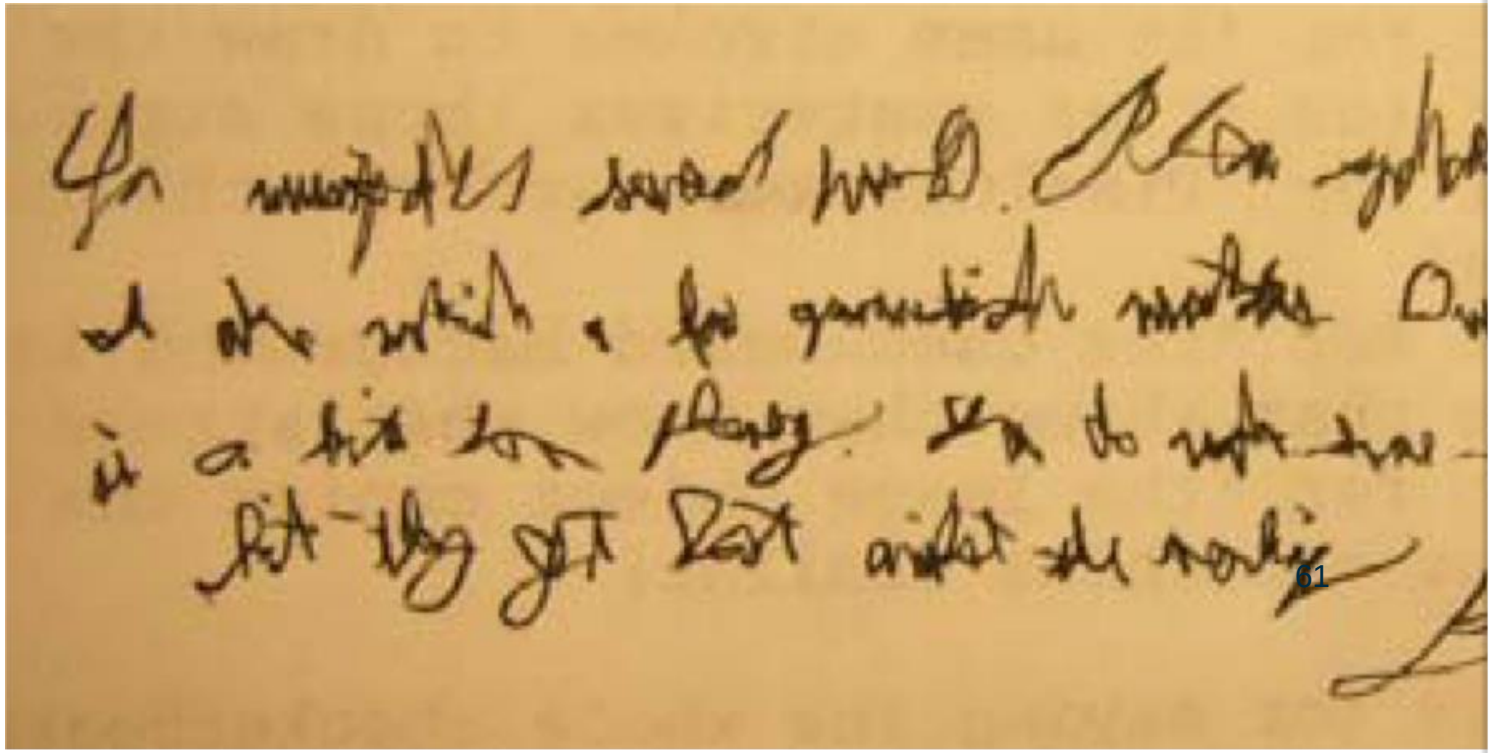


Improve



Verify and vote

Crowdsourcing workflow examples: Iterative



version 1:

You (?) (?) (?) (work). (?) (?) (?) work (not) (time). I (?) (?) a few grammatical mistakes. Overall your writing style is a bit too (phoney). You do (?) have good (points), but they got lost amidst the (writing). (signature)

version 4:

You (misspelled) (several) (words). (?) (?) (?) work next (time). I also notice a few grammatical mistakes. …

version 5:

You (misspelled) (several) (words). (Plan?) (spellcheck) (your) work next time. I also notice a few grammatical mistakes. Overall your writing style is a bit too phoney. You do make some good (points), but they got lost amidst the (writing). (signature)

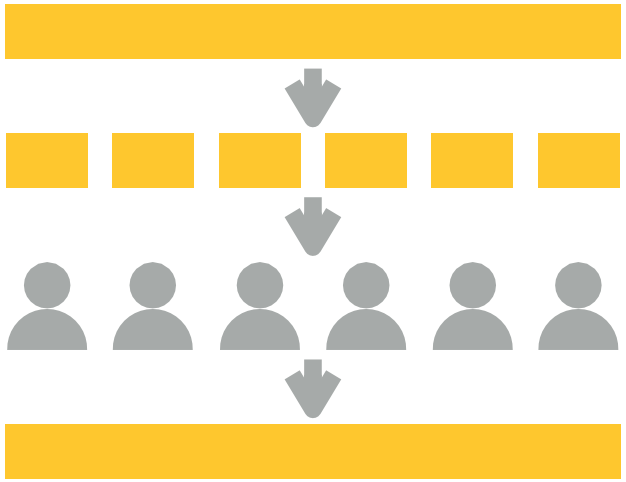
version 6:

You (misspelled) (several) (words). Please spellcheck your work next time. I also notice a few grammatical mistakes. Overall your writing style is a bit too phoney. You do make some good (points), but they got lost amidst the (writing). (signature)

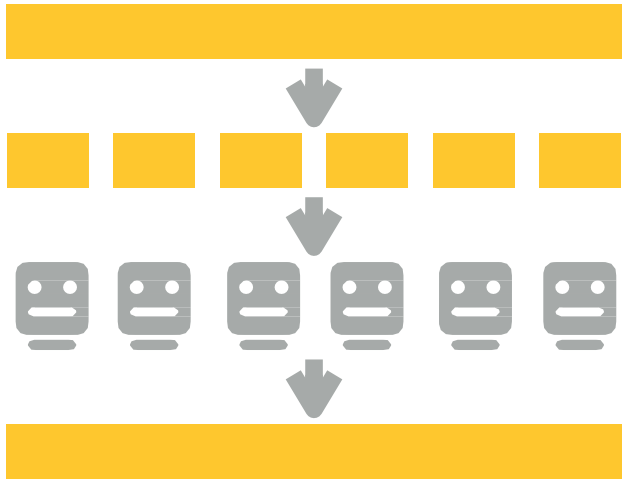
You (misspelled) (several) (words). Please spellcheck your work next time. I also notice a few grammatical mistakes. Overall your writing style is a bit too phoney. You do make some good (points), but they got lost amidst the (writing). (signature)

Crowdsourcing workflow vs. Chaining in LLMs

Crowdsourcing workflow



LLM chain



Similarities…

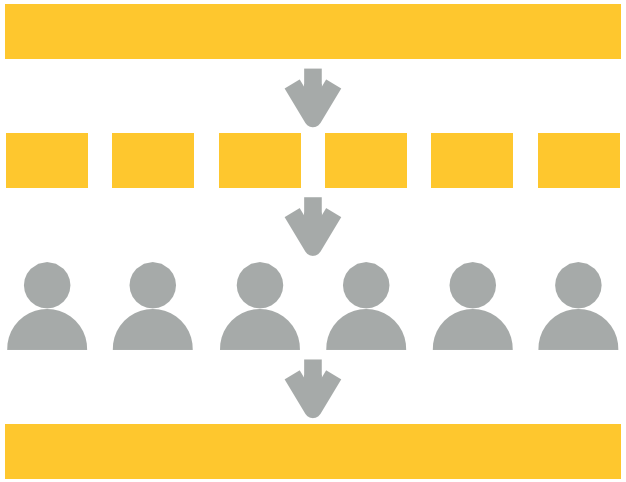
Idea: breakdown complex tasks into pieces that can be done independently, then combined.

Gains: scale to tasks that are otherwise hard, more structured interactions, more resilient to interruptions (of distractor tasks).

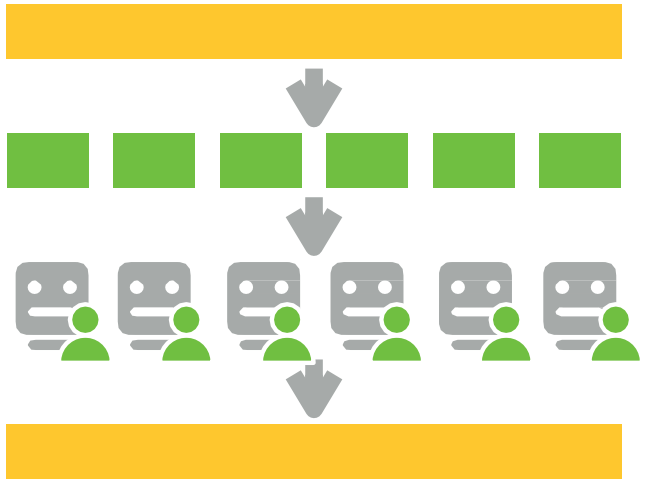
Limitations: cascading errors, conflicts between parallel paths, etc.

Crowdsourcing workflow vs. Chaining in LLMs

Crowdsourcing workflow



LLM chain



Differences…

Breakdown rational

Address pitfalls of a single worker
Can do any task,
But do tasks with high variance,
Unwilling to digest too much context

Address pitfalls of a single LLM pass
Has intensive computing power,
But limited reasoning capability,
have exposure bias, etc.

Human access

Humans only have access to the steps assigned to them
More chance of conflict

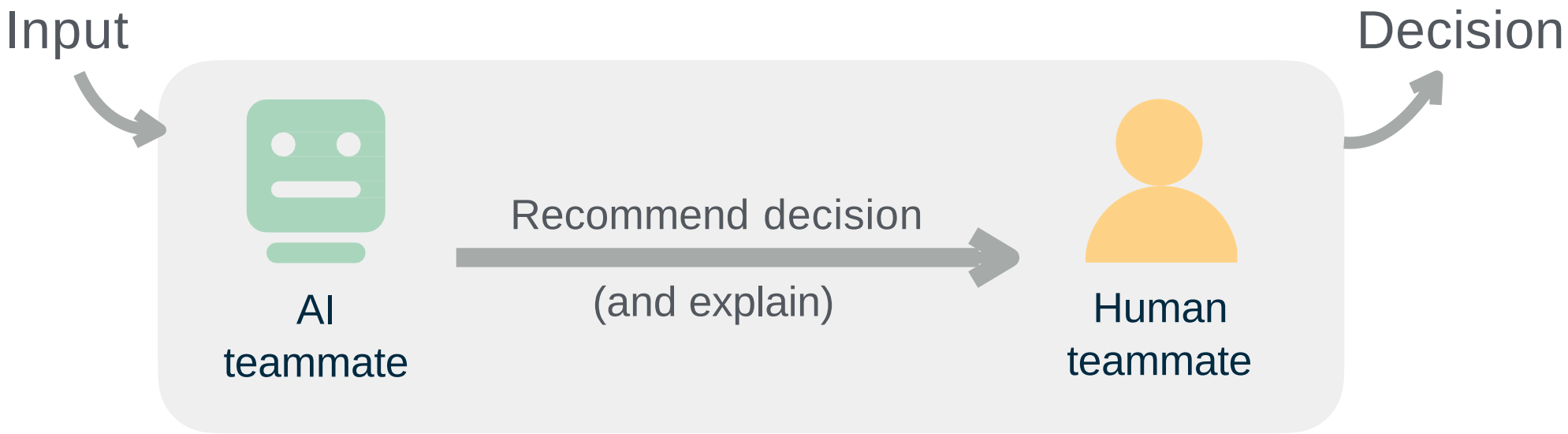
Humans may interrupt at any step
Can take advantage of cascading effects, parallel paths, for explainability

Human-AI Collaboration: Key Aspects

- **Complementary performance**: Leverage the strengths of both AI and humans, to achieve better outcomes than either could accomplish alone.
- **Alignment**: AI should behave in a ways that humans expect their “teammate” to behave.
- **Cooperative interaction**: The interaction between human and AI should work in a way that’s both intuitive for the human, and compatible to model characteristics.
- **Reasonable task allocation**: Assign tasks that the model can perform.
- **Appropriate trust**: humans should follow AIs when they are correct, but identify and correct their mistakes otherwise.

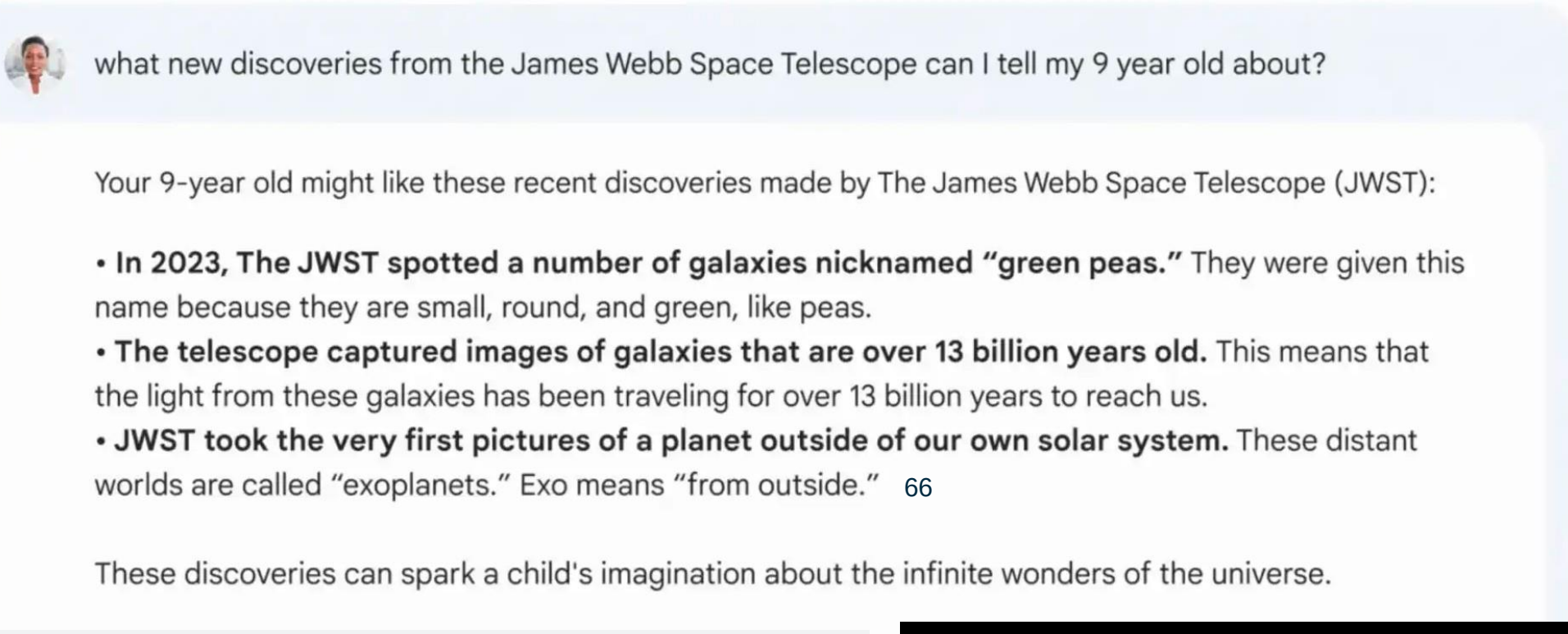
Appropriate trust in HAI team decision making

Human and AI perform the same task; AI makes suggested decisions and maybe explain, and human makes the final decision



Appropriate trust: ***accept*** *AI recommendations when they are correct, but* ***overwrite*** *them when they don't make sense.*

Appropriate reliance is hard.



Which fact is wrong?

"first pictures of exoplanets were taken by the European Southern Observatory's Very Large Telescope (VLT) in 2004, as confirmed by NASA."

Design interactions for appropriate reliance

One example: When & how to display the AI recommendation matters.

inline highlight ➜ confirmation bias (“this sentence seems reasonable enough”)

See AI decision first ➜ anchoring effect (“I will agree with AI’s decision”)

HCI solution

Display of AI recommendation

Present evidence, but not final decision; provide more explanations

Timing of AI decision

Asynchronous display, increase independence

Park, Joon Sung, et al. "A Slow Algorithm Improves Users' Assessments of the Algorithm's Accuracy." CSCW 2019

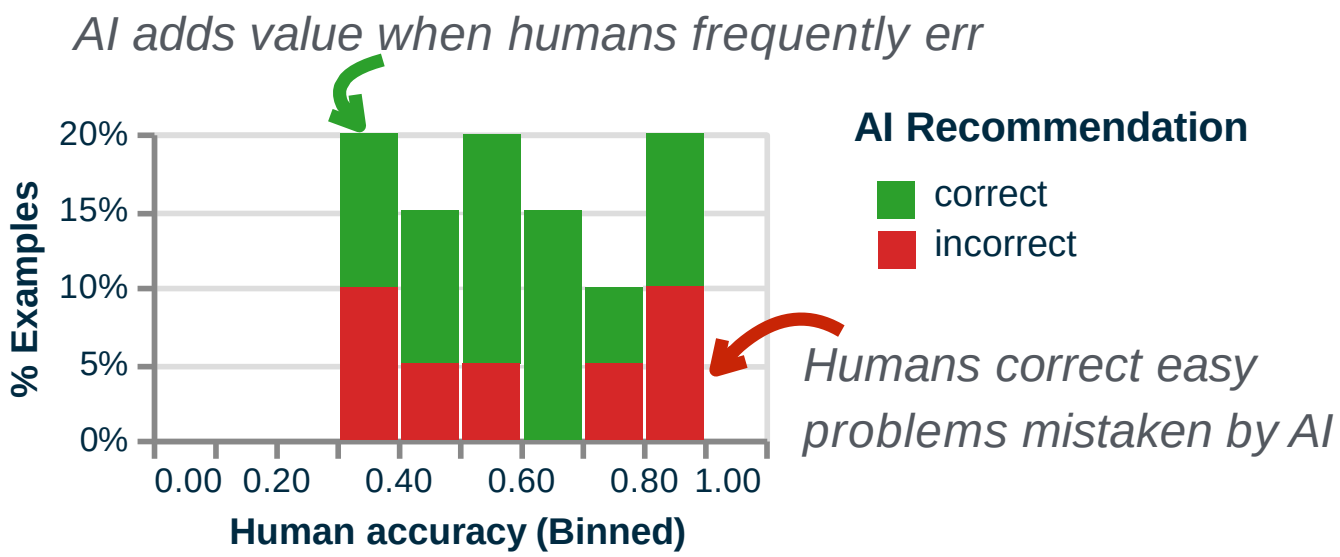
Wang, Danding, et al. "Designing theory-driven user-centric explainable AI." CHI 2019

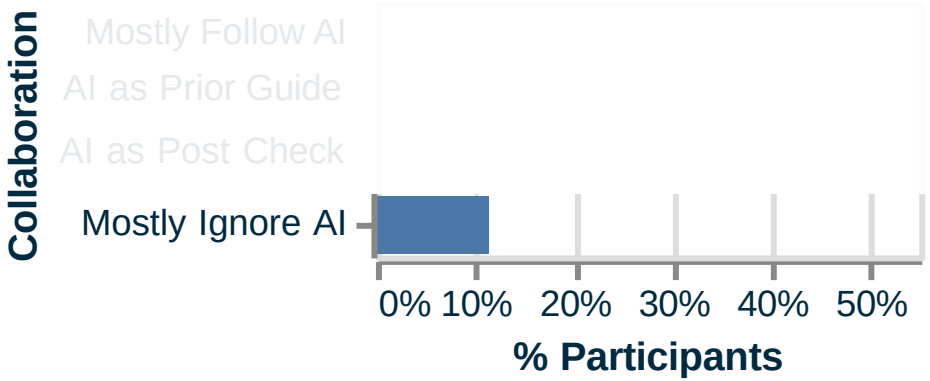
Factors causing inappropriate reliance?

The distribution of human and AI expertise matters.

Human+AI is ineffective if they make the same kinds of mistakes.

Multi-choice question answering task



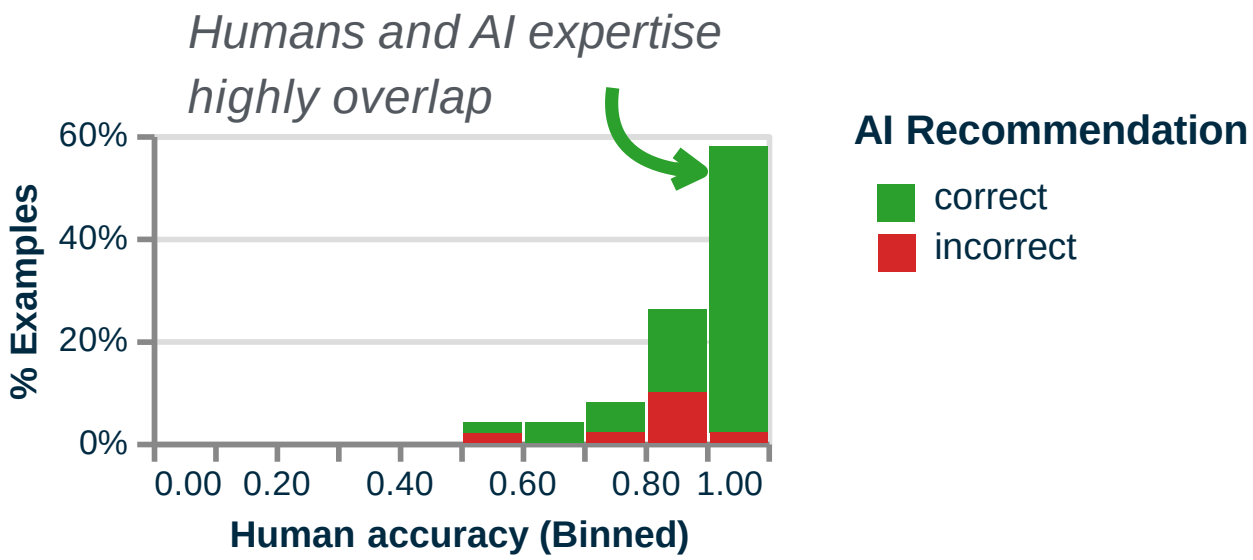


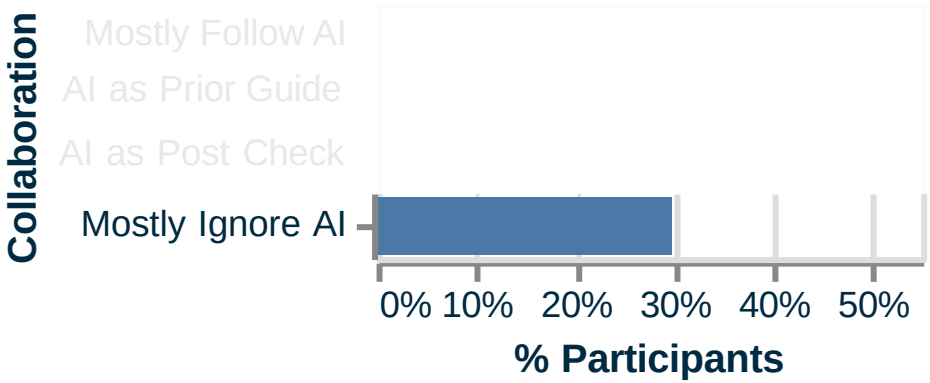
Factors causing inappropriate reliance?

The distribution of human and AI expertise matters.

Human+AI is ineffective if they make the same kinds of mistakes.

Classification task





Choose AIs carefully for the task

The distribution of human and AI expertise matters.

Human+AI is ineffective if they make the same kinds of mistakes.

AI solution

Change the training objective
Directly optimize for complementary behavior

HCI solution

Re-think AI's roles, to help in other dimensions
speed, cognitive load, etc.

Bansal, Gagan, et al. "Is the Most Accurate AI the Best Teammate? Optimizing AI for Teamwork." AAAI 2021
Feng, Shi, and Jordan Boyd-Graber. "What can ai do for me? evaluating machine learning interpretations in cooperative play." IUI 2019.
Glassman, Elena L., et al. "OverCode: Visualizing variation in student solutions to programming problems at scale." TOCHI 2015

Summary

- Humans interact with models in different ways.
- Key factors of Human-model Interaction includes:
    - **Complementary performance.**
    - **Alignment.**
    - **Reasonable task allocation.**
    - **Cooperative interaction.**
    - **Appropriate trust.**

Prompting is cool but needs to consider model capabilities.

Questions?