Interactive Fiction and Text Generation

Lara J. Martin (she/they)

https://laramartin.net/interactive-fiction-class

Learning Objectives

Consider when to use various sampling algorithms

Distinguish between finetuning and prompting

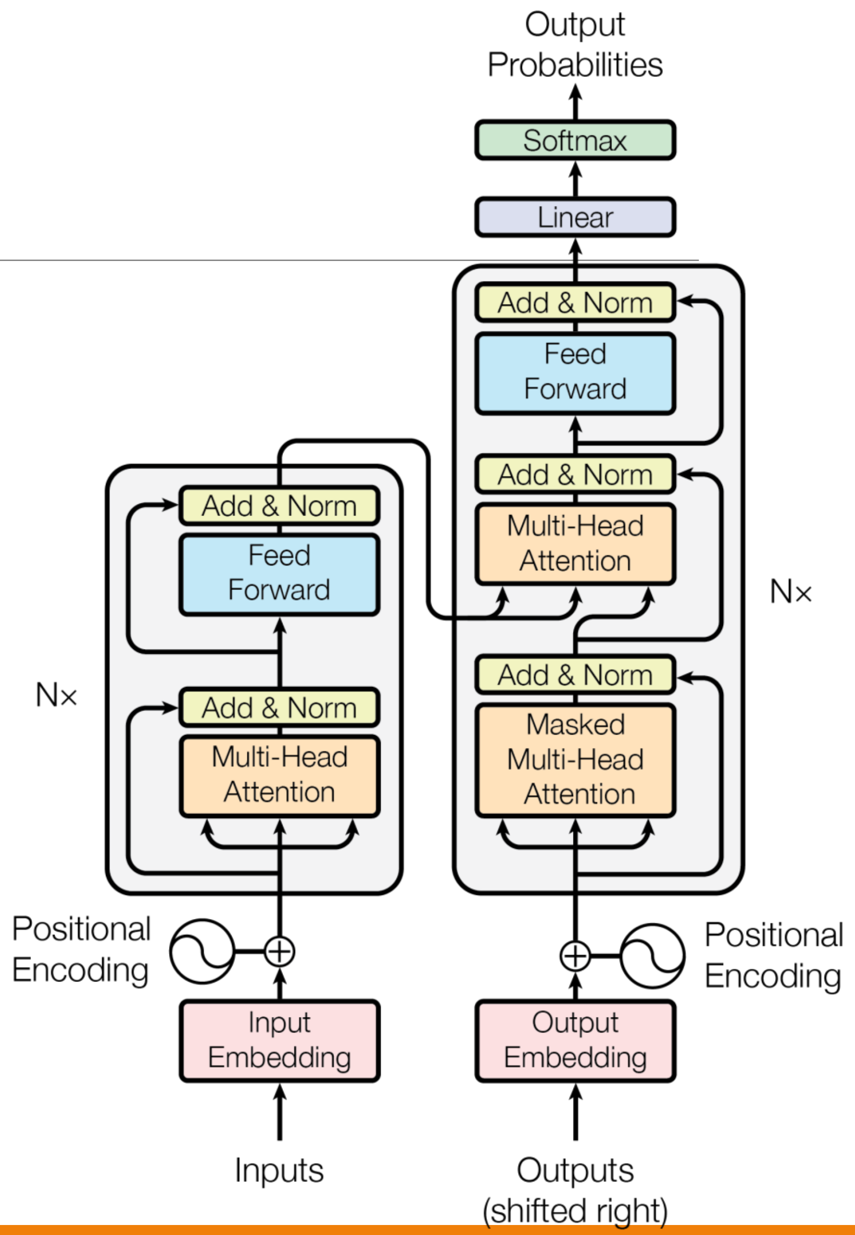
Distinguish between few-shot and zero-shot prompting

Examine the ways GPT’s parameters affect sampling

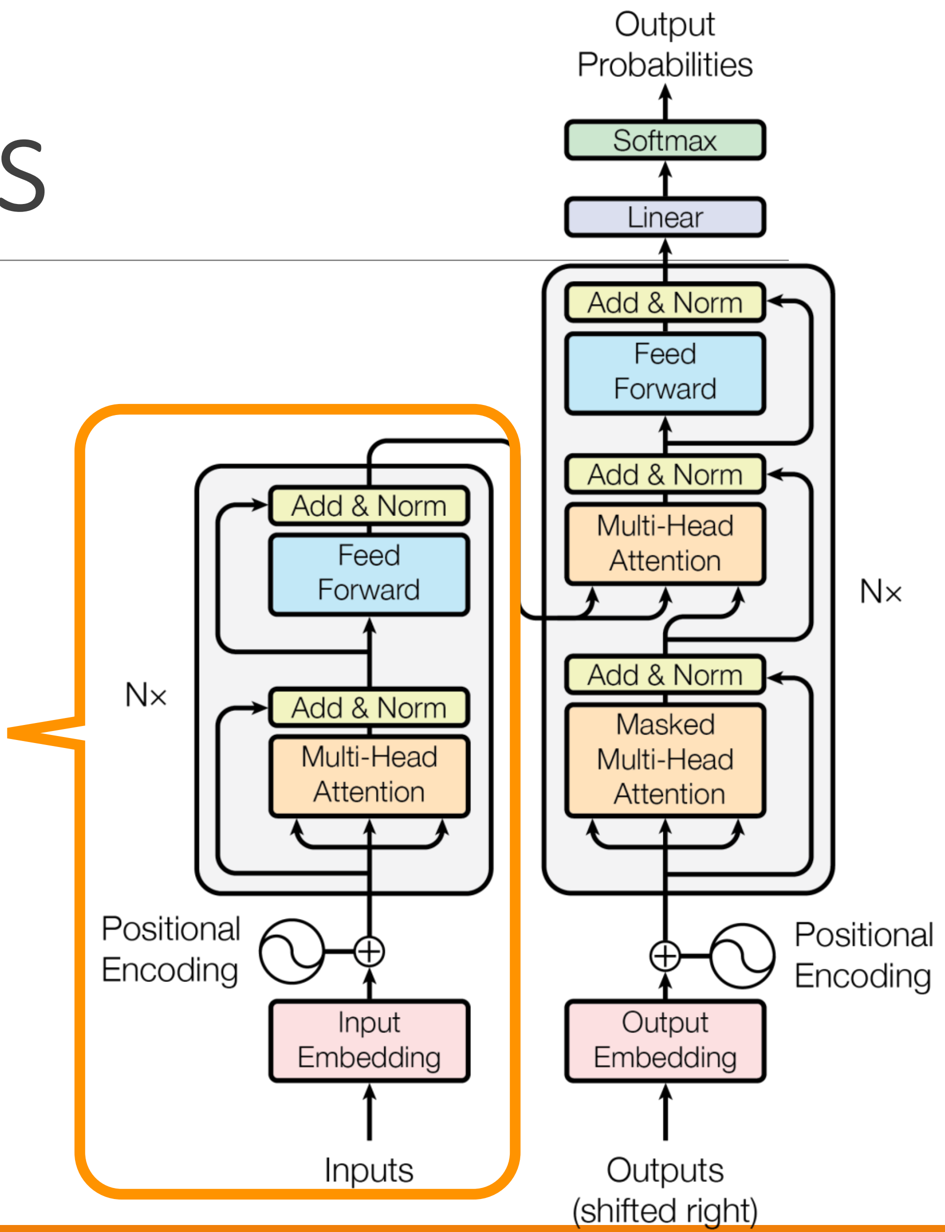
Review: Transformers

The Transformer is a **non-recurrent** non-convolutional (feed-forward) neural network designed for language understanding

- introduces self-attention in addition to encoder-decoder attention

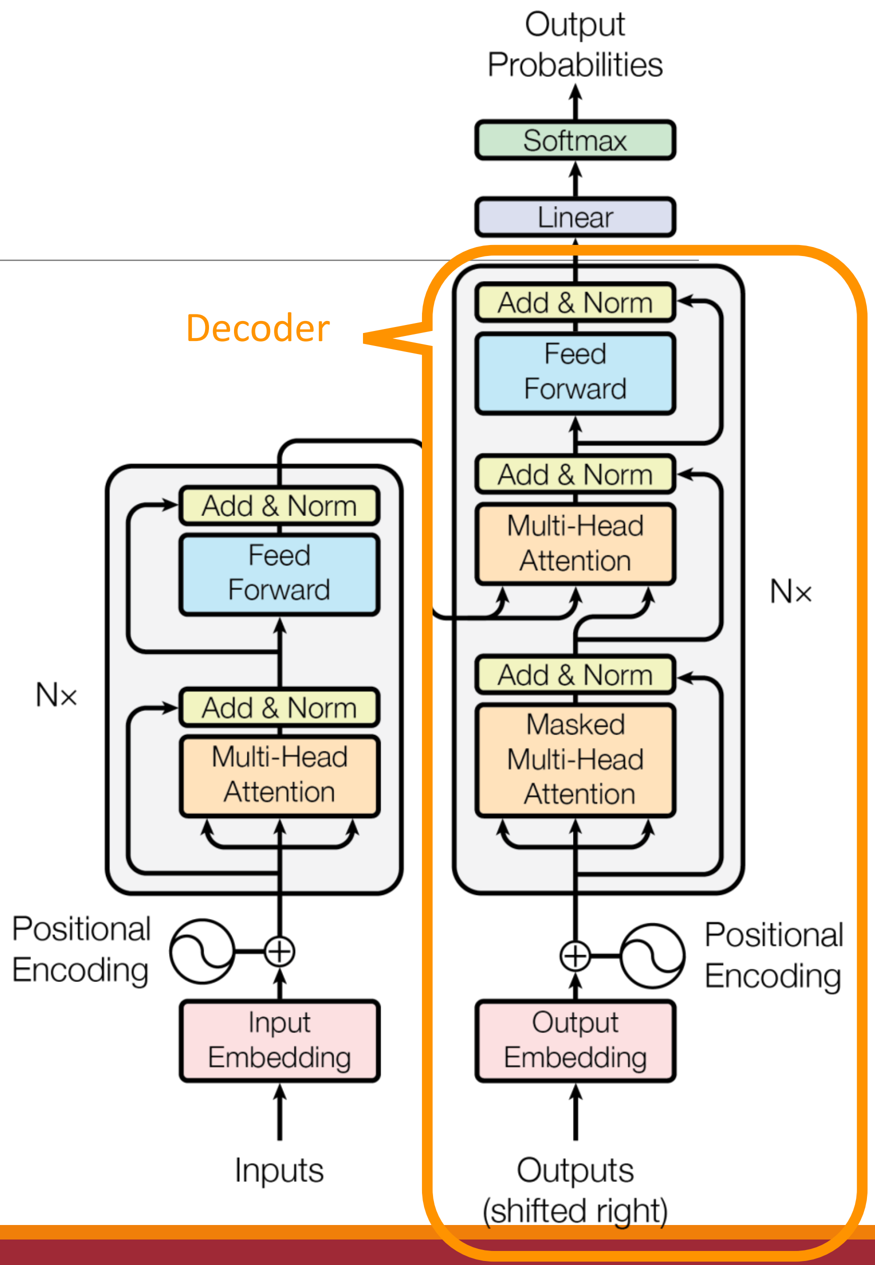


Review: Transformers

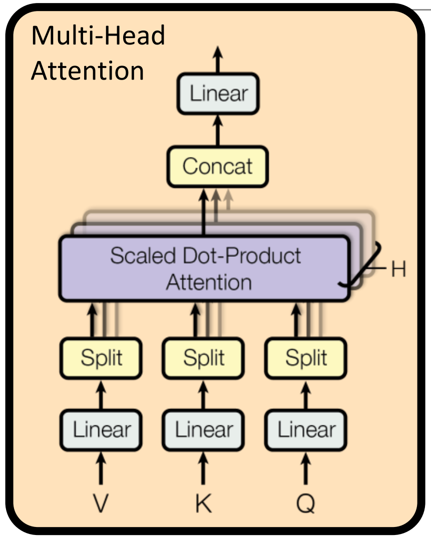


Encoder

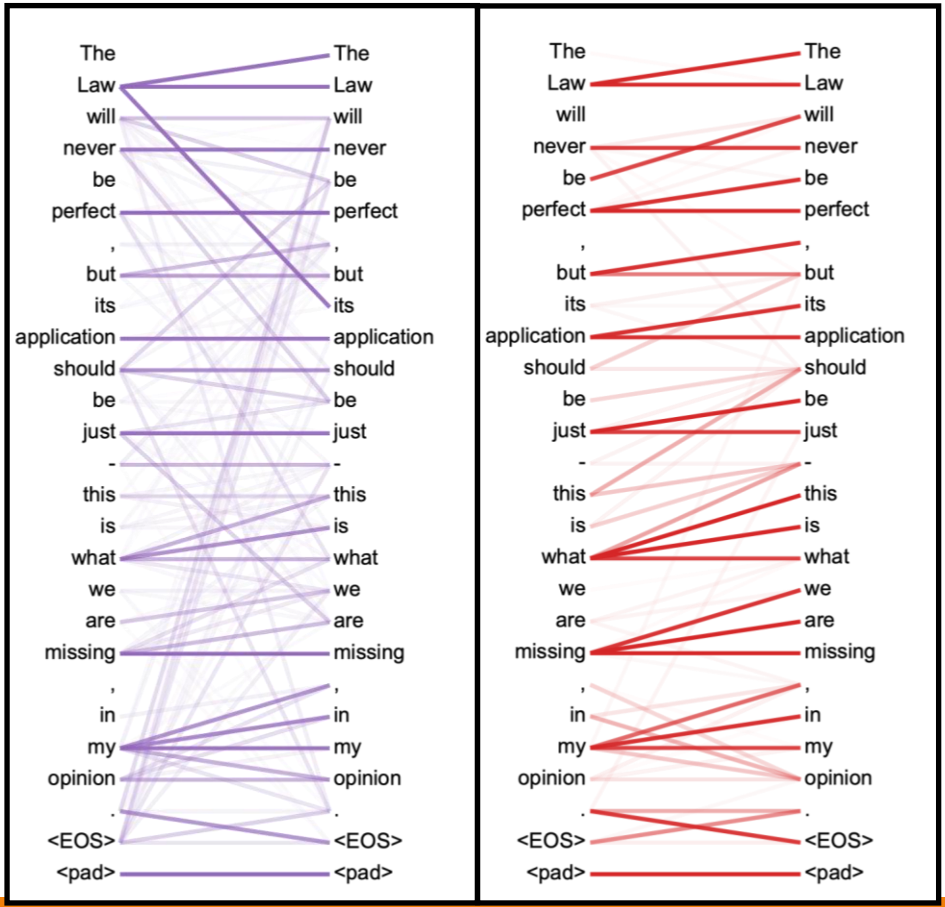
Review: Transformers

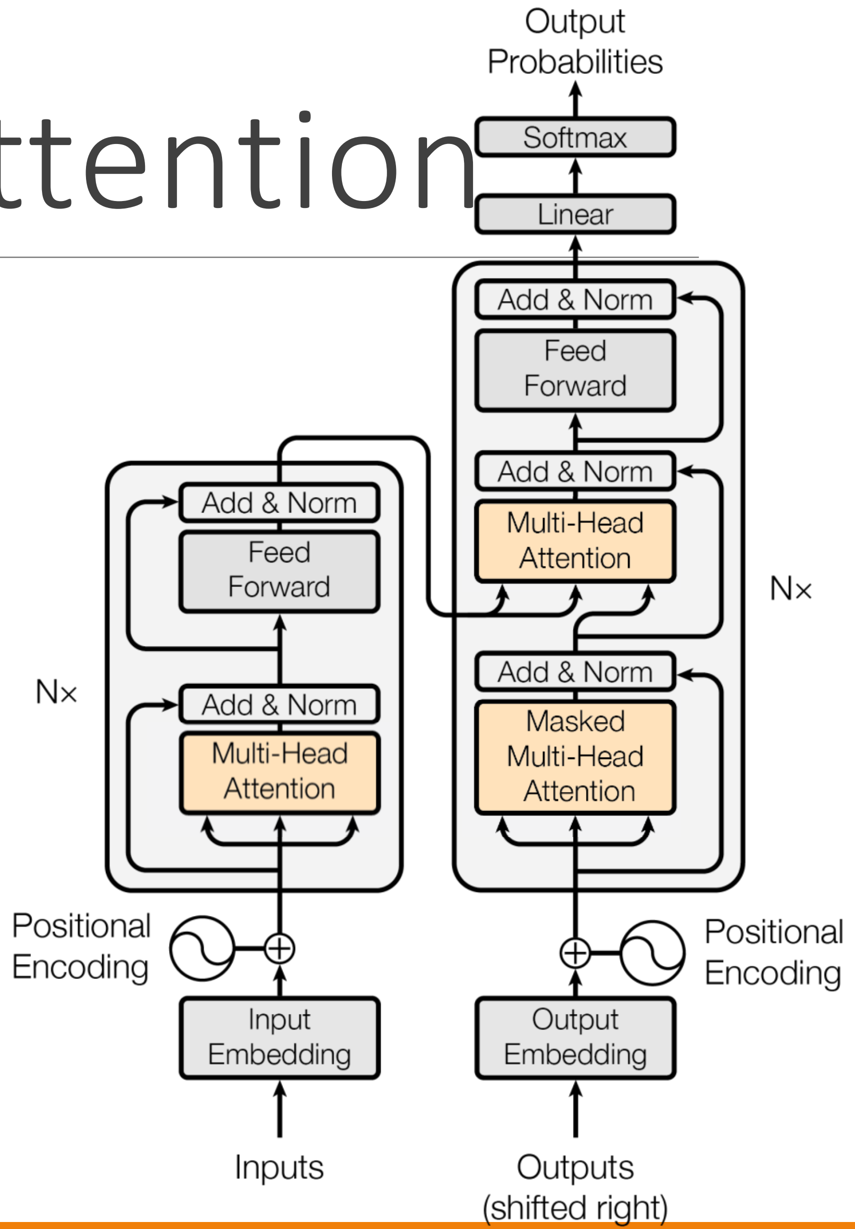


Review: Multi-Head Attention



Two different self-attention heads:





Review: Strengths of the Transformer Architecture

Training is easily parallelizable

- Larger models can be trained efficiently.

Does not “forget” information from earlier in the sequence.

- Any position can attend to any position.

Review: Weaknesses of the Transformer Architecture

We can use a lot of data to train → expensive (money, time)

Can’t actually remember things, just looks back

Review: Generating Text

Also sometimes called decoding 🙄

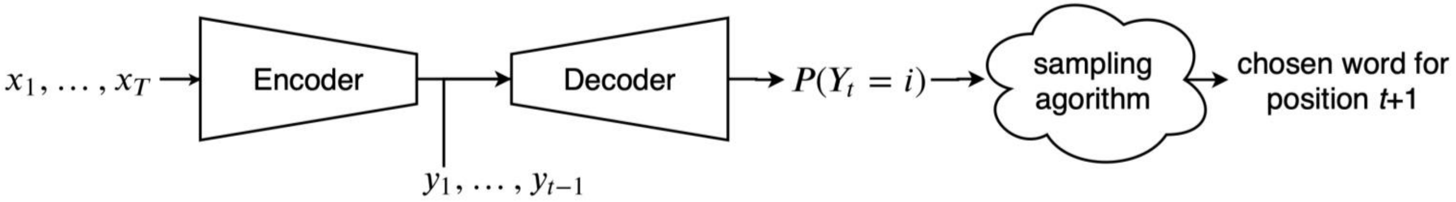
To generate text, we need an algorithm that selects tokens given the predicted probability distributions.

Examples:

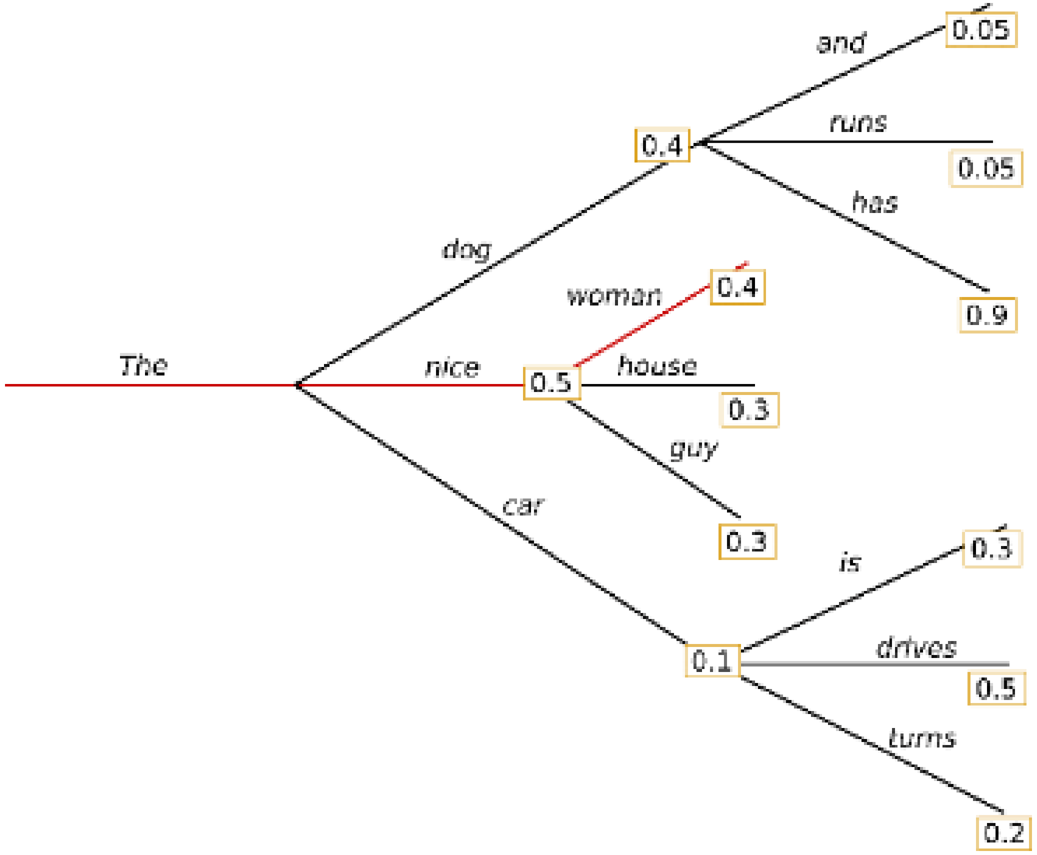
Argmax

Random sampling

Beam search



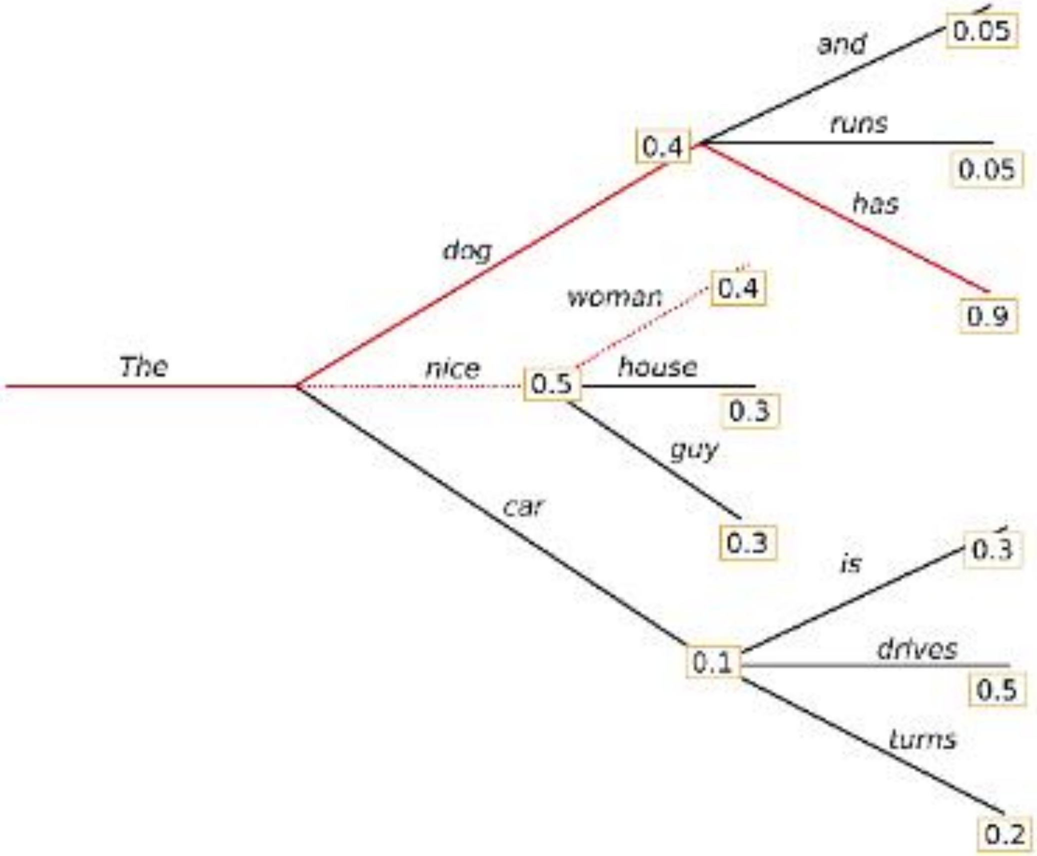
Greedy Search (Argmax)



https://huggingface.co/blog/how-to-generate

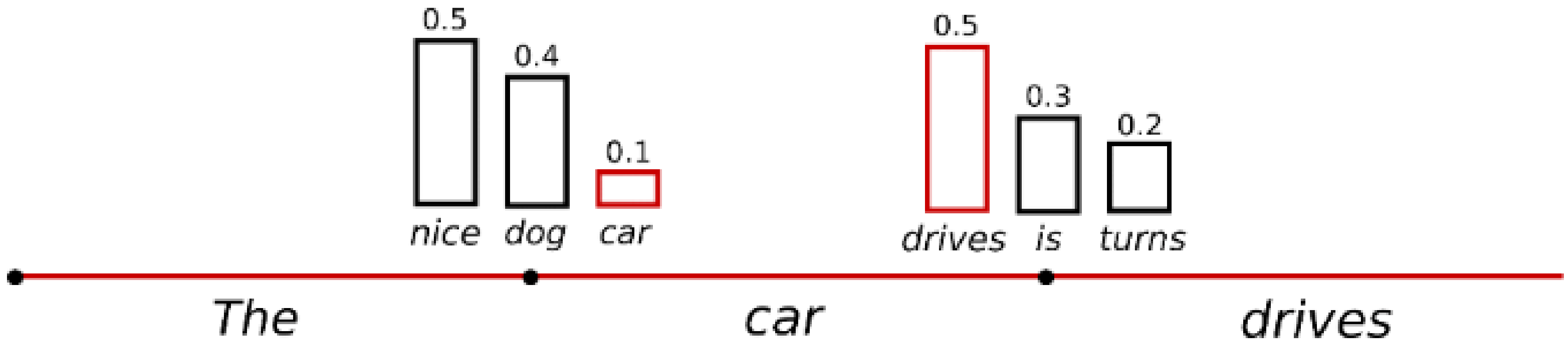
Beam Search

Number of
beams = 2



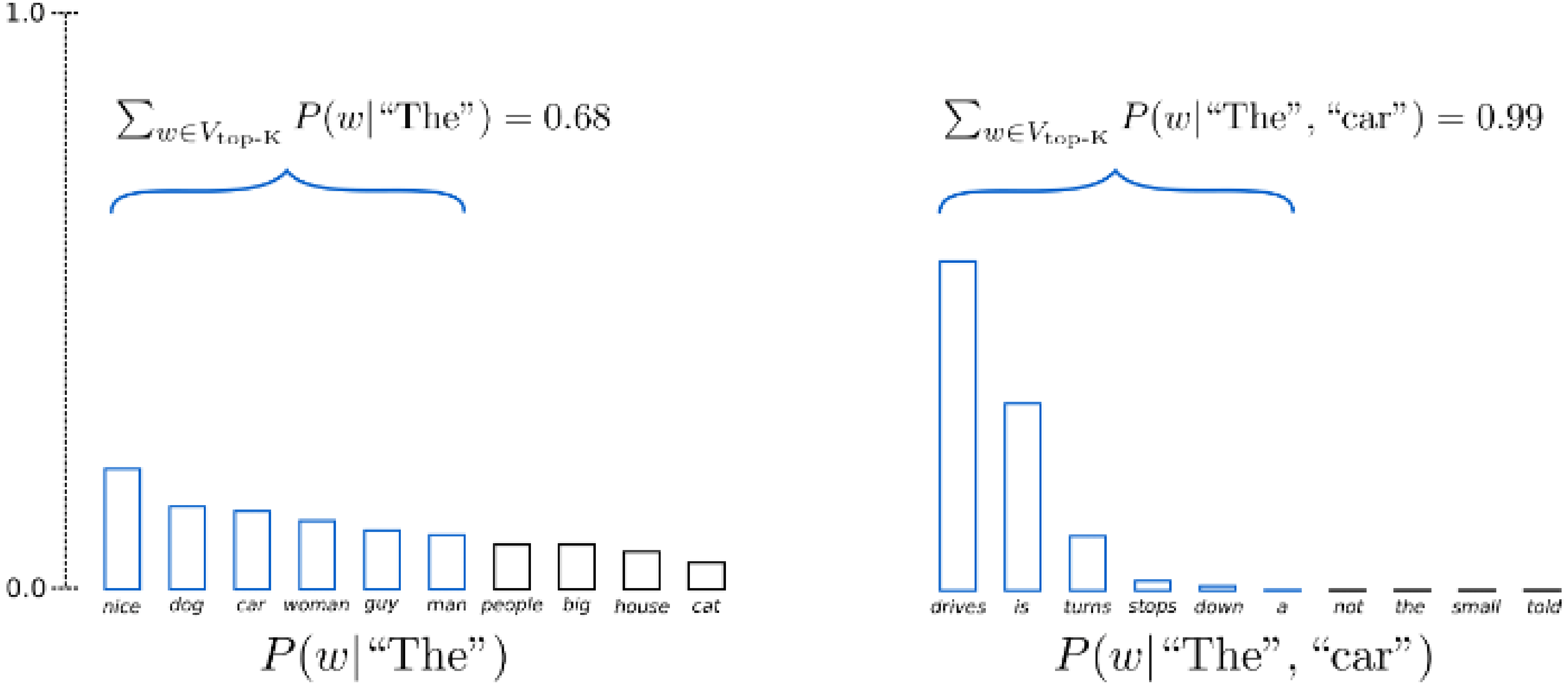
https://huggingface.co/blog/how-to-generate

Random Sampling



https://huggingface.co/blog/how-to-generate

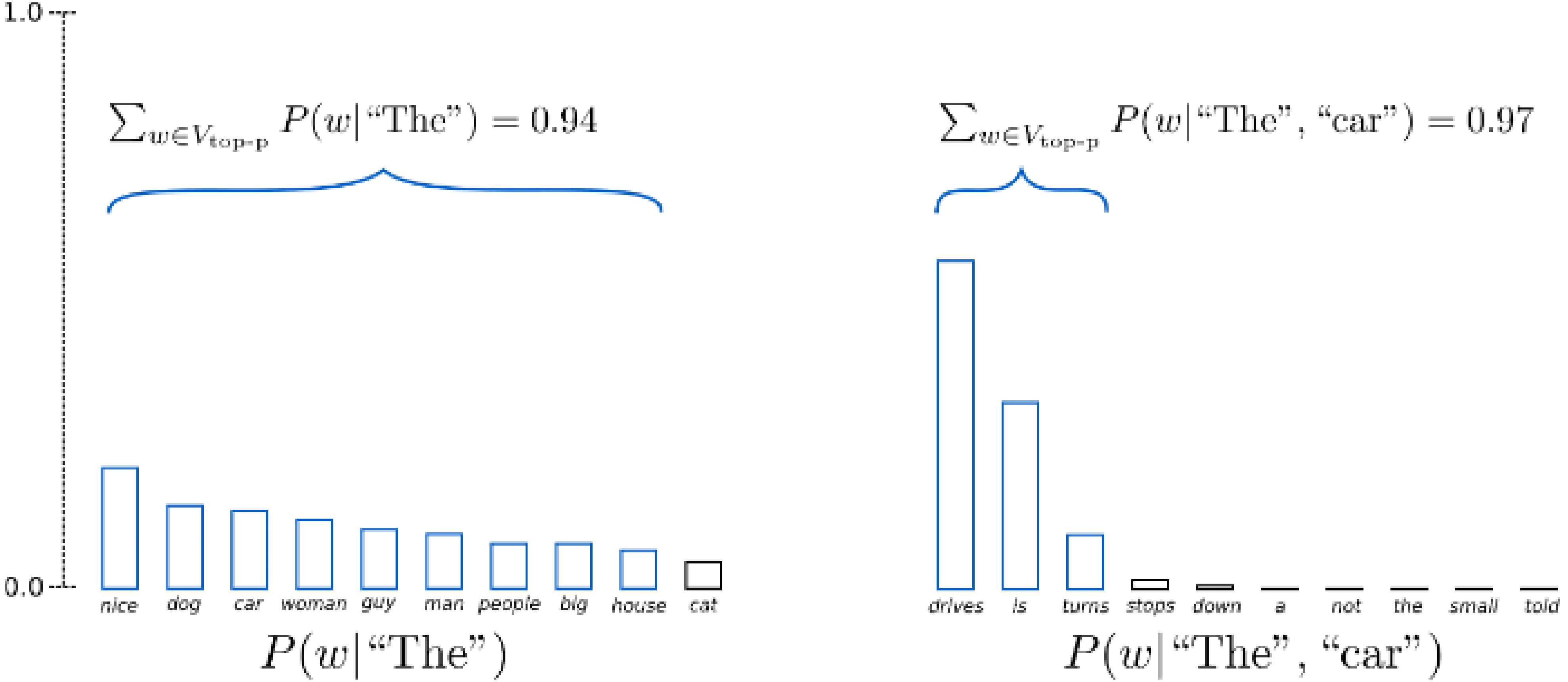
Top-K Sampling



A. Holtzman, J. Buys, M. Forbes, and Y. Choi, "The Curious Case of Neural Text Degeneration," in *International Conference on Learning Representations (ICLR)*, 2020, p. 16. https://openreview.net/forum?id=rygGQyrFvH

https://huggingface.co/blog/how-to-generate

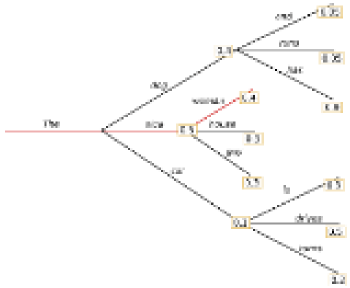
Top-P Sampling



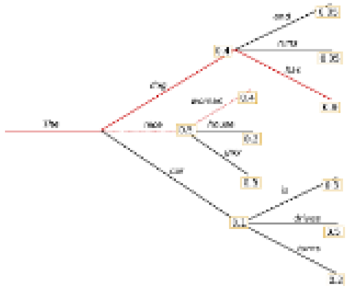
https://huggingface.co/blog/how-to-generate

Think-Pair-Share

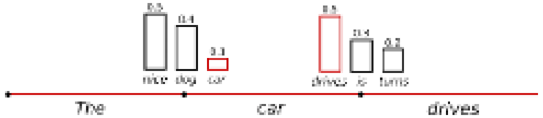
When might you want to use one sampling algorithm over the other?



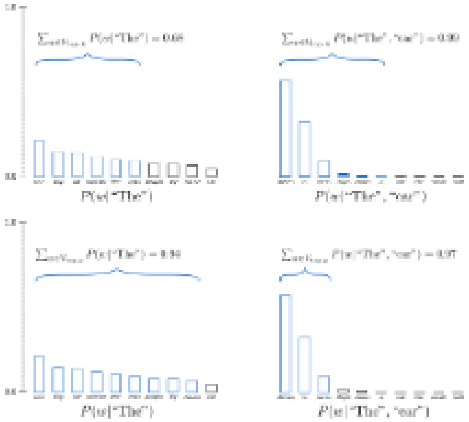
Greedy



Beam Search

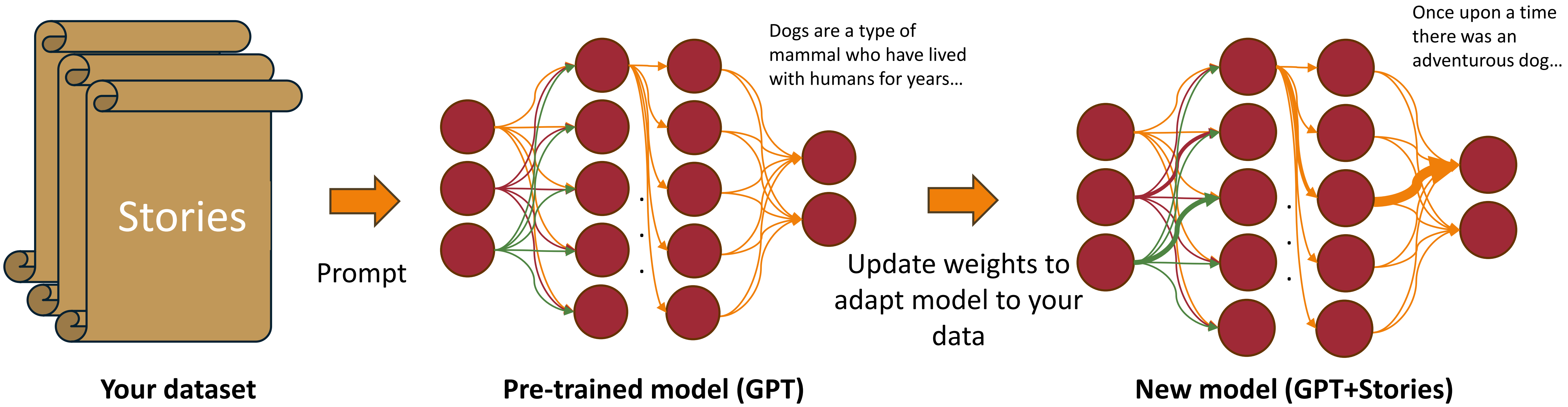


Random Sampling



Top-K/P

Finetuning



Prompt

Dogs are a type of mammal who have lived with humans for years…

Update weights to adapt model to your data

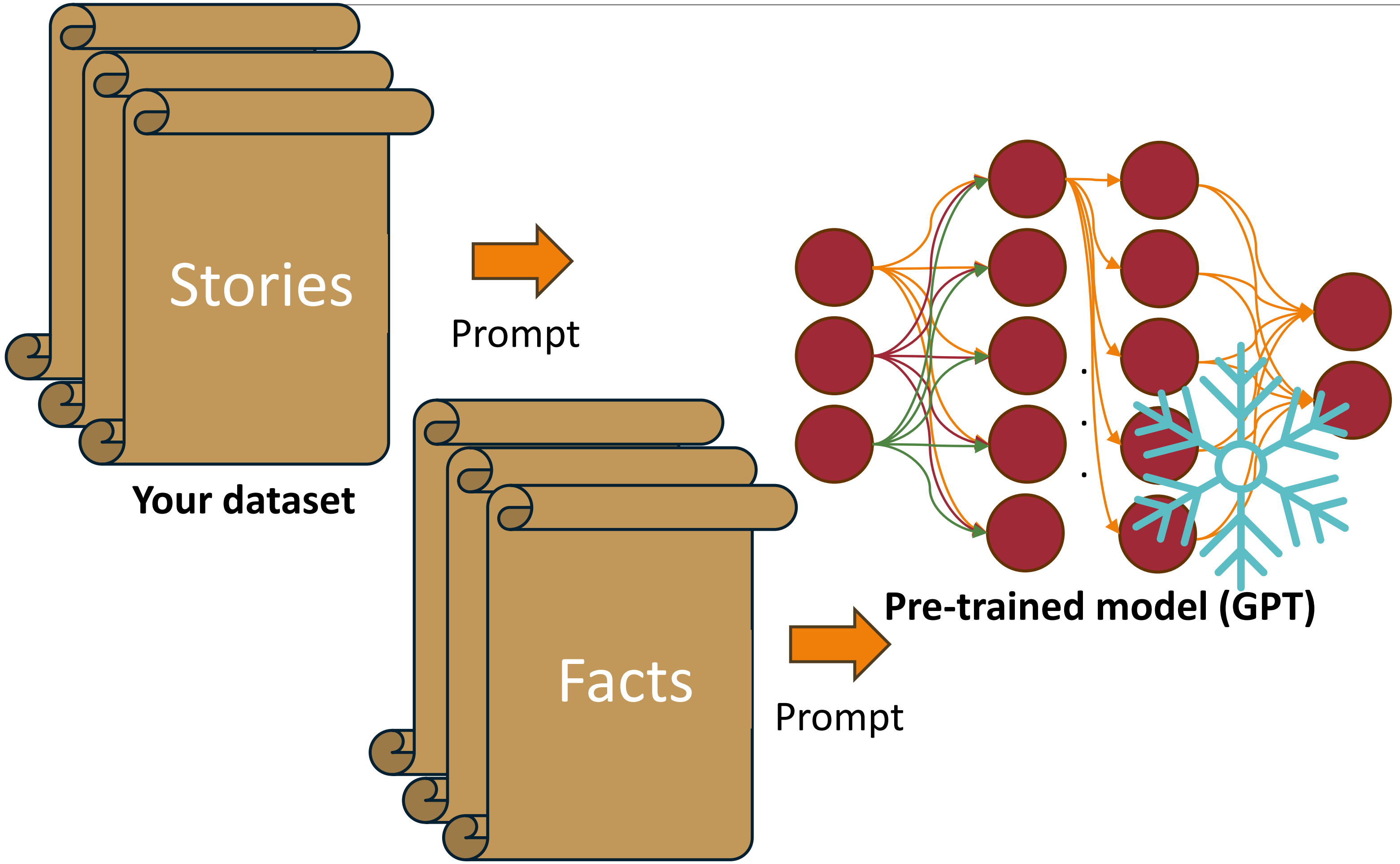
Once upon a time there was an adventurous dog…

Your dataset

Pre-trained model (GPT)

New model (GPT+Stories)

Prompting



Stories

Prompt

Your dataset

Facts

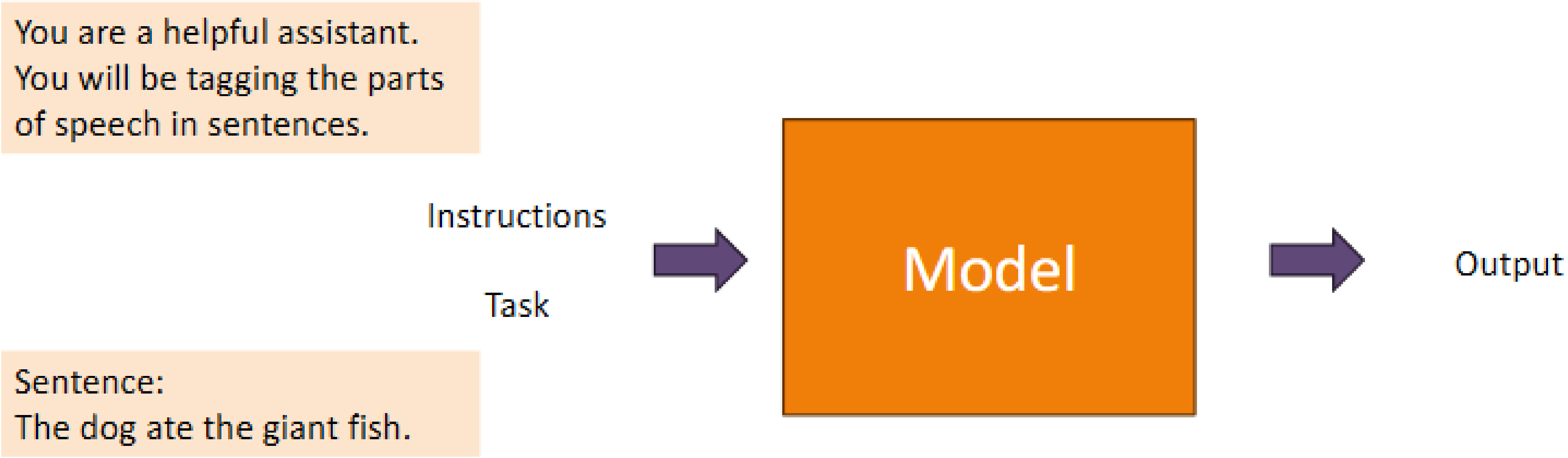
Prompt

Pre-trained model (GPT)

Once upon a time there was an adventurous dog…

Dogs are a type of mammal who have lived with humans for years…

Zero-Shot Prompting



You are a helpful assistant. You will be tagging the parts of speech in sentences.

Sentence:
The dog ate the giant fish.

Few-shot Prompting

Instructions

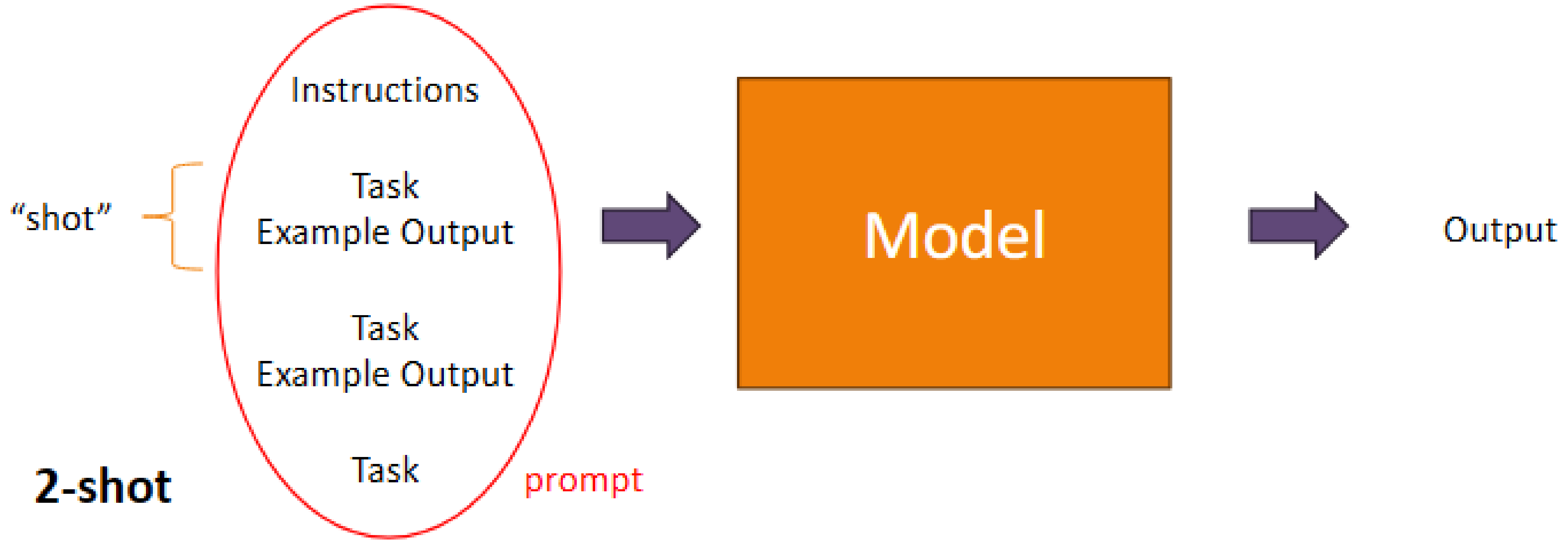
You are a helpful assistant.
You will be tagging the parts of speech in sentences.

Task

Sentence:
The dog ate the giant fish.

Example Output

The dog ate the giant fish.
D N V D Adj N



Instructions

"shot"

Task
Example Output

Task
Example Output

Task

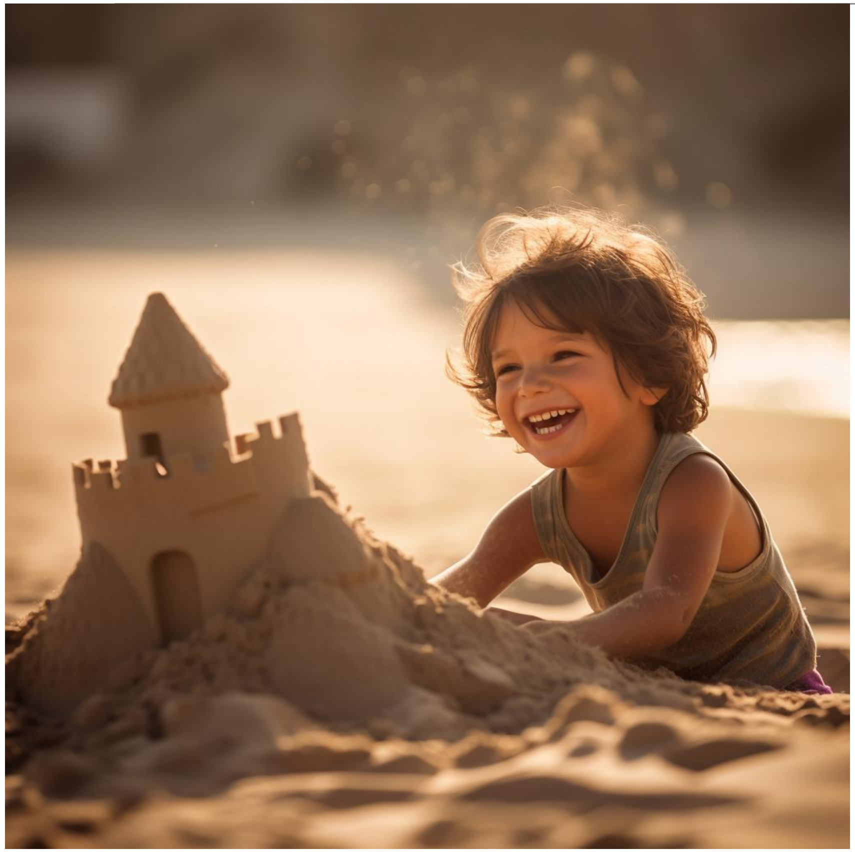
prompt

Model

Output

2-shot

Prompting



"A child playing on a sunny happy beach, their laughter as they build a simple sandcastle, emulate Nikon D6 high shutter speed action shot, soft yellow lighting."
Generated with Midjourney.
via https://zapier.com/blog/ai-art-prompts/

Need to be really specific
(also match the training data)

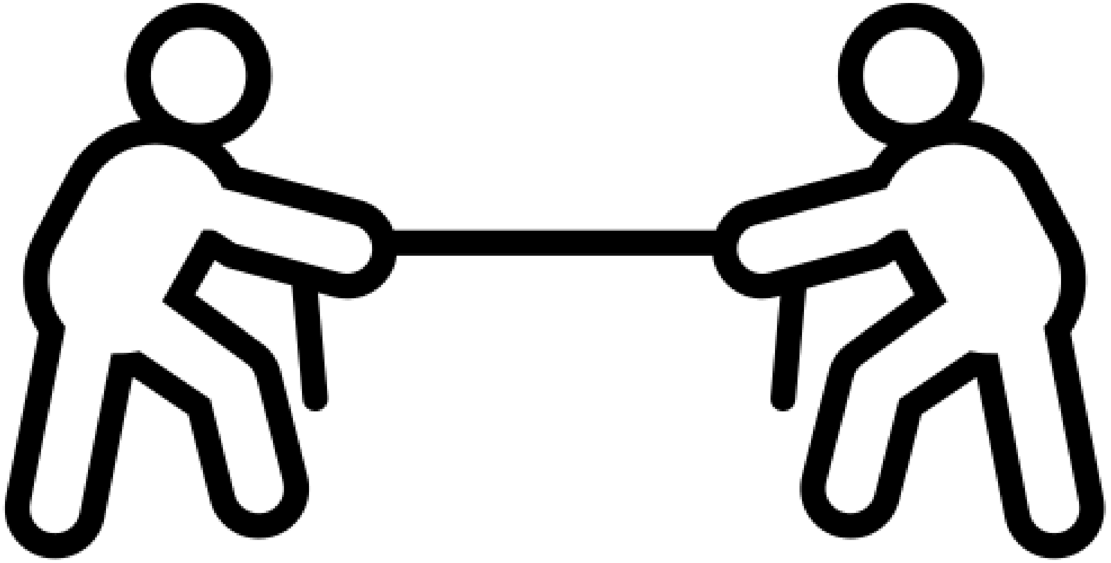
Dealing with any language model

Likelihoods → Not cause & effect

What is probable might not be possible.

Lara’s Language Model Tradeoff

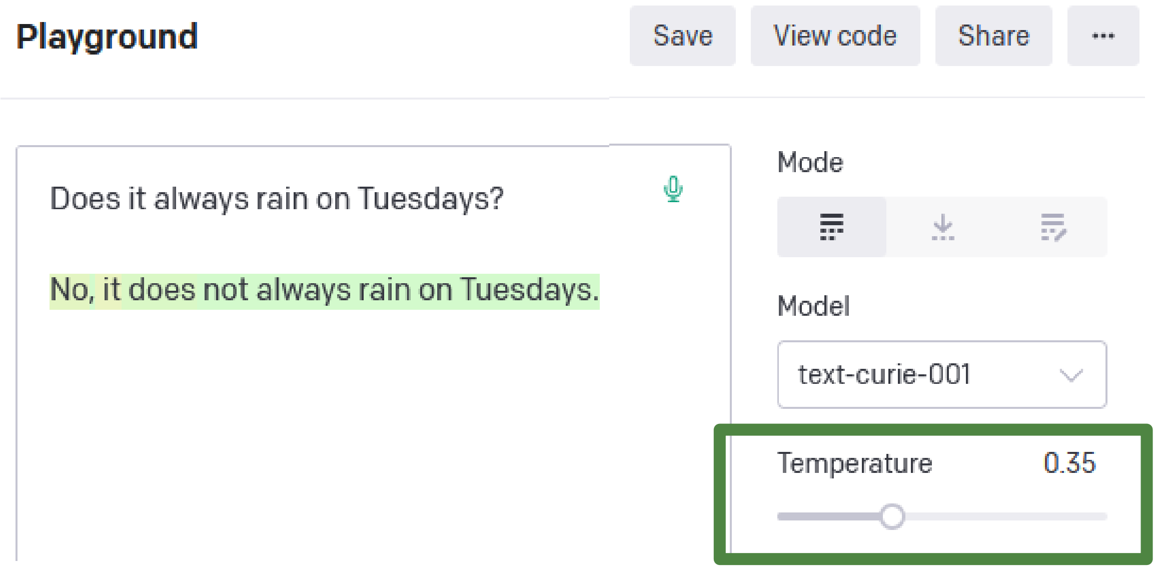
Coherence



Originality

https://thenounproject.com/icon/tug-of-war-1016981/

There’s even an explicit knob in GPT



Does it always rain on Tuesdays?

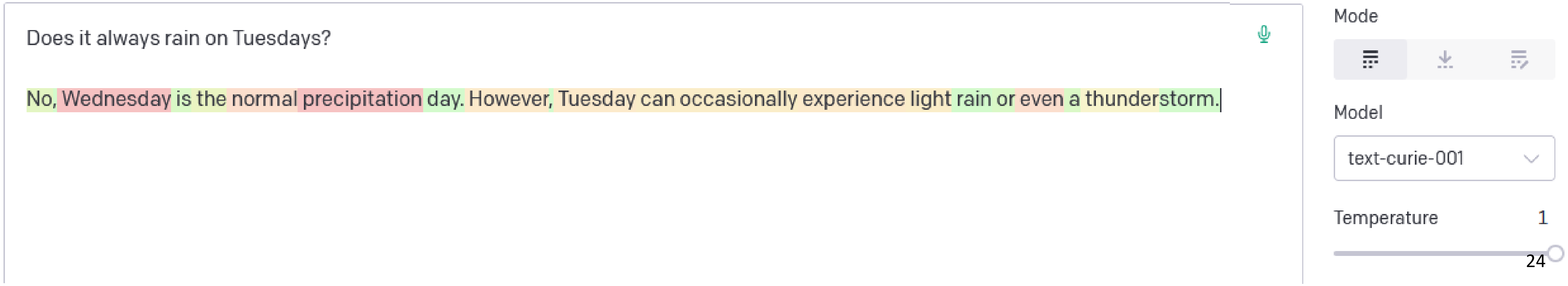
No, it does not always rain on Tuesdays.

Mode

Model

text-curie-001

Temperature 0.35


Does it always rain on Tuesdays?

No, Wednesday is the normal precipitation day. However, Tuesday can occasionally experience light rain or even a thunderstorm.

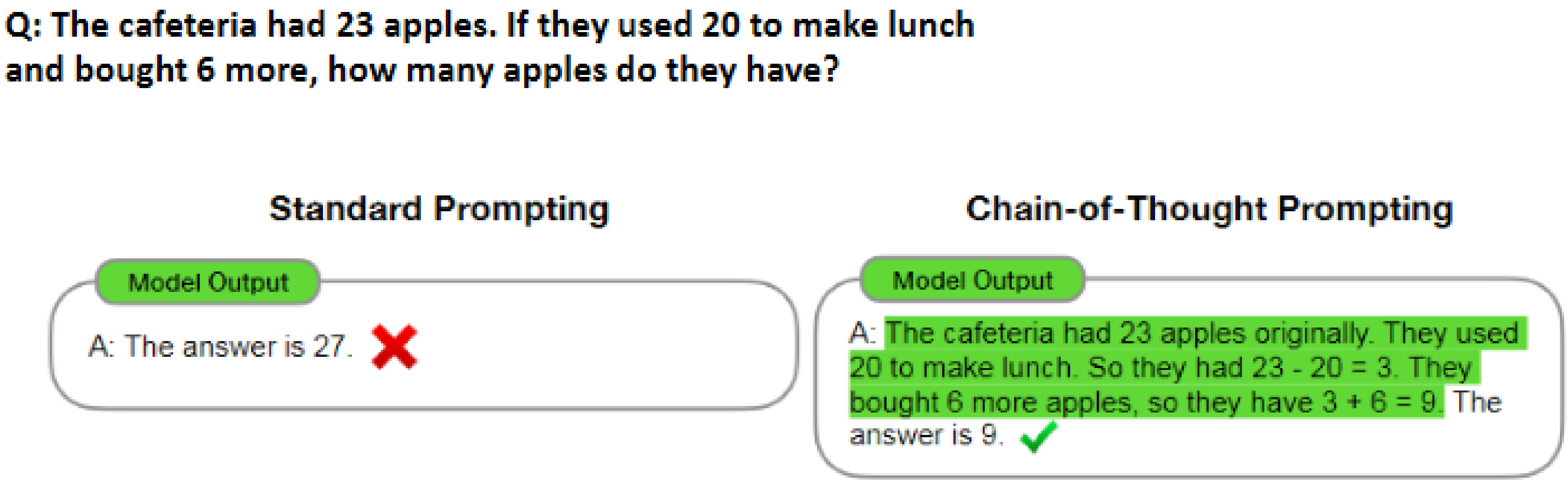
Mode

Model

text-curie-001

Temperature 1

Chain-of-Thought Prompting



Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Standard Prompting

Model Output

A: The answer is 27. ✗

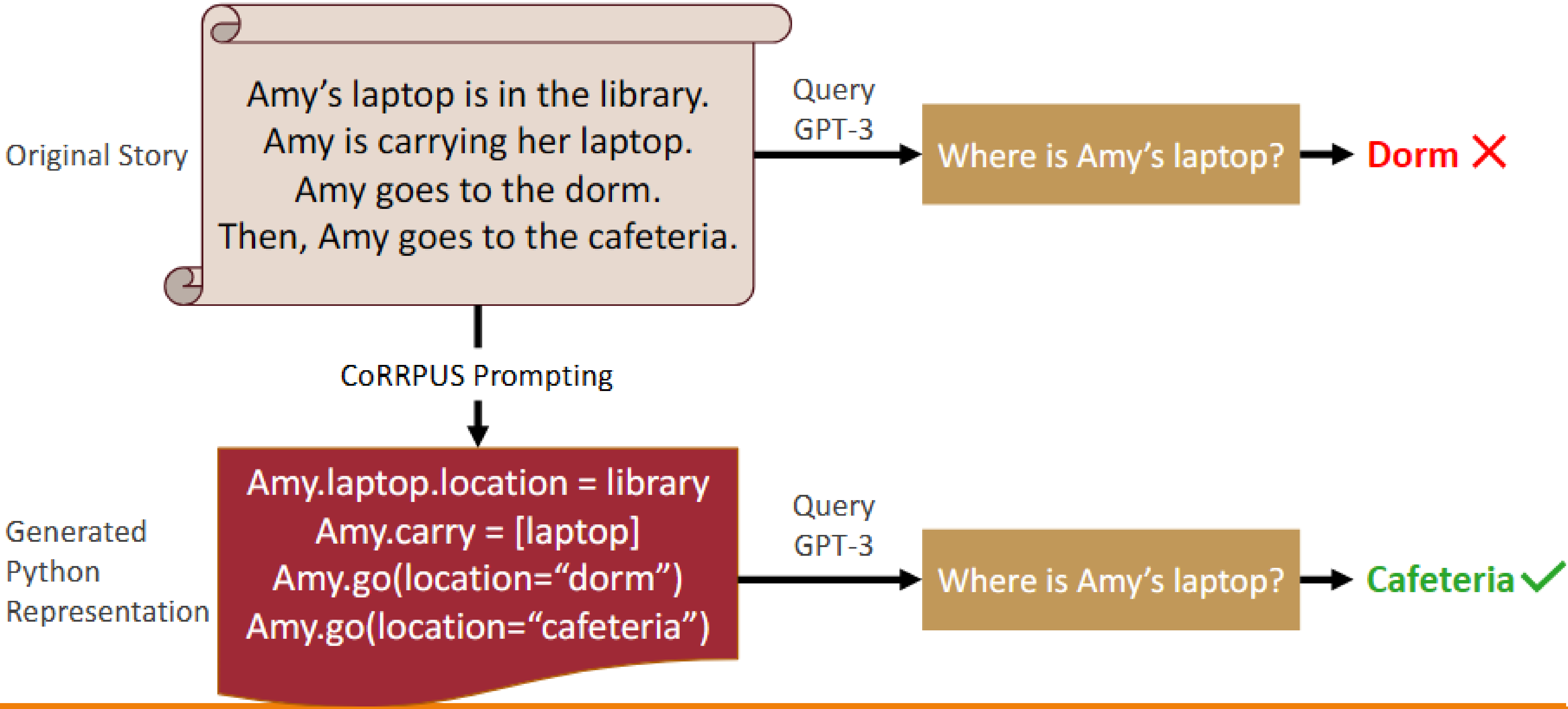
Chain-of-Thought Prompting

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Part of Figure 1 from J. Wei et al., "Chain of Thought Prompting Elicits Reasoning in Large Language Models,"
in International Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA & Online, Jun. 2022. doi: 10.48550/arXiv.2201.11903.

CoRRPUS (Code Representations to Reason & Prompt over for Understanding in Stories)



Original Story

Amy’s laptop is in the library.
Amy is carrying her laptop.
Amy goes to the dorm.
Then, Amy goes to the cafeteria.

Query GPT-3

Where is Amy’s laptop? → Dorm ✗

CoRRPUS Prompting

Generated Python Representation

```
Amy.laptop.location = library
Amy.carry = [laptop]
Amy.go(location=“dorm”)
Amy.go(location=“cafeteria”)
```

Query GPT-3

Where is Amy’s laptop? → Cafeteria ✓

Dong, Y. R., Martin, L. J., & Callison-Burch, C.
“CoRRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding.” Findings of ACL 2023.

CoRRPUS Chain-of-Thought Prompting

Three versions that are initialized the same:

Comment

```
def story(self):
        ## Mary moved to the bathroom.
        self.Mary.location = “bathroom”
        ## Mary got the football there.
        self.Mary.inventory.append(“football”)
        …
```

Specific Functions

```
self.Mary_moved_to_the_bathroom()
self.Mary_got_the_football_there()
self.John_went_to_the_kitchen()
self.Mary_went_back_to_the_garden()

def Mary_moved_to_the_bathroom()
        self.Mary.location=“bathroom”
def Mary_got_the_football_there():
…
```

Abstract Functions

```
def go(self, character, location):
        character.location = location
        for item in character.inventory:
                item.location = location
def pick_up(): …

def story(self):
        ## Mary moved to the bathroom.
        self.go(character=self.Mary,
        location = “bathroom”)
        …
```

Dong, Y. R., Martin, L. J., & Callison-Burch, C.
“CoRRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding.” Findings of ACL 2023.

Tested on 2 Tasks

bAbI (Weston et al. 2015)

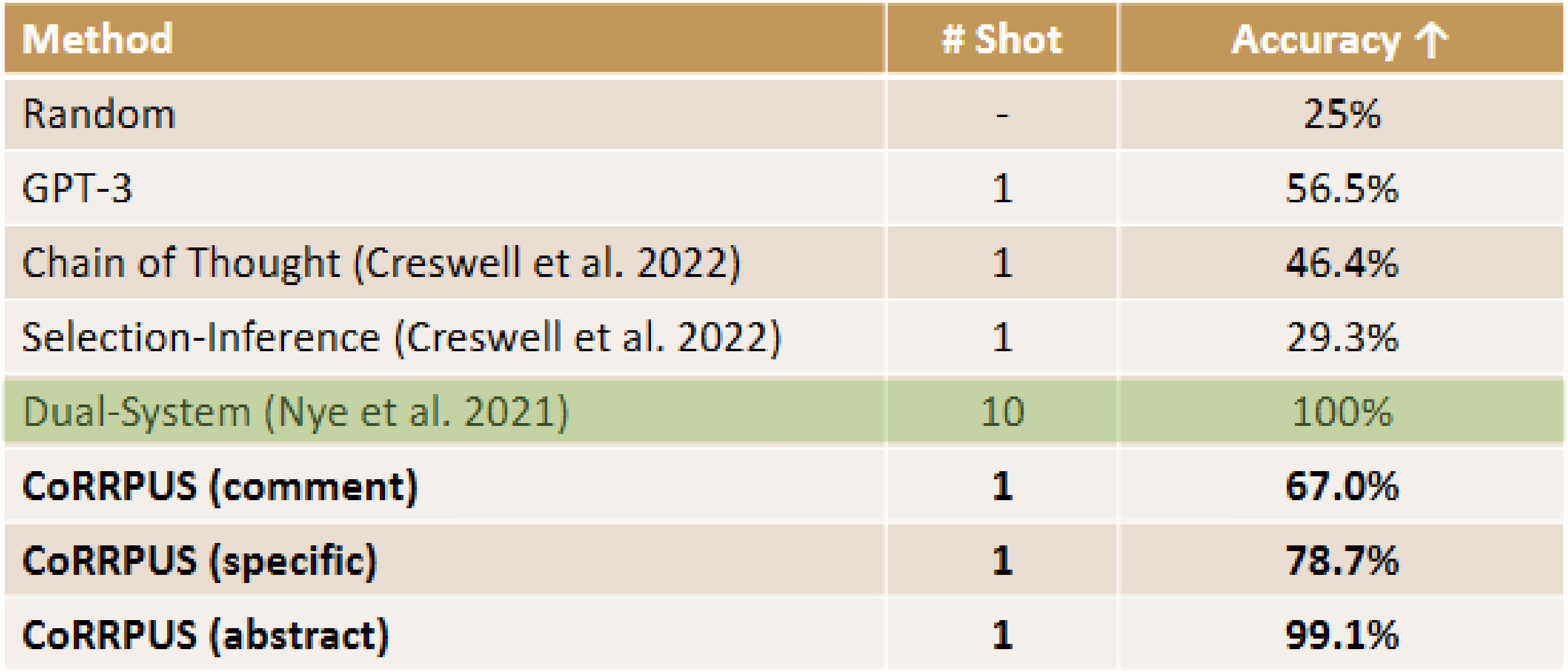
- Task 2: Stories tracking objects that characters carry

Re3 (Yang et al. 2022)

- Identifying inconsistencies in stories (e.g., descriptions of characters' appearances, relationships)
- Stories were generated from a list of facts (the premise). They also generated premises with a contradiction.

Dong, Y. R., Martin, L. J., & Callison-Burch, C.
"CoRRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding." Findings of ACL 2023.

bAbI (Weston et al. 2015)

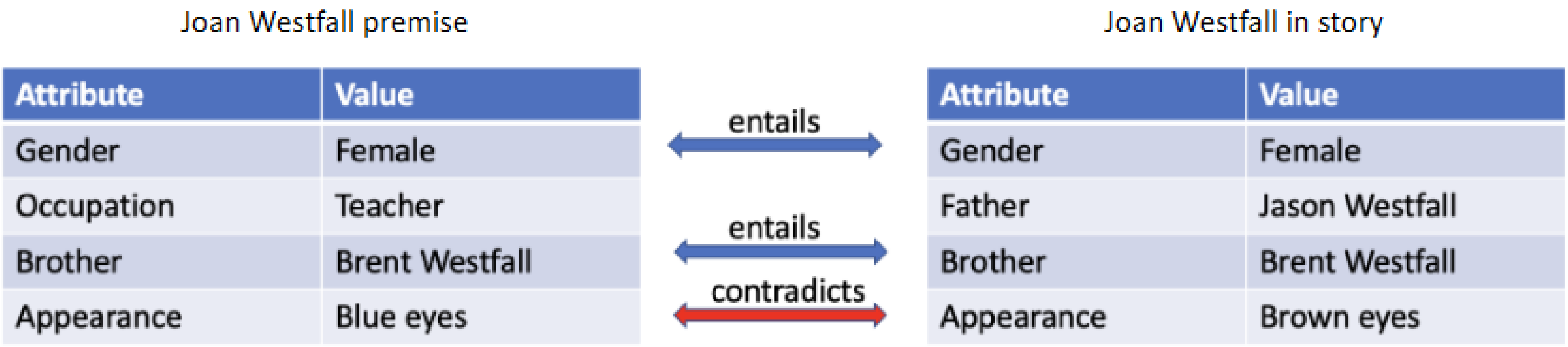


Method	# Shot	Accuracy ↑
Random	-	25%
GPT-3	1	56.5%
Chain of Thought (Creswell et al. 2022)	1	46.4%
Selection-Inference (Creswell et al. 2022)	1	29.3%
Dual-System (Nye et al. 2021)	10	100%
CoRRPUS (comment)	**1**	**67.0%**
CoRRPUS (specific)	**1**	**78.7%**
CoRRPUS (abstract)	**1**	**99.1%**

Dong, Y. R., Martin, L. J., & Callison-Burch, C.
“CoRRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding.” Findings of ACL 2023.

Re^3

The task is to see what stories match what premises based on the facts extracted from both.



Joan Westfall premise

Attribute	Value
Gender	Female
Occupation	Teacher
Brother	Brent Westfall
Appearance	Blue eyes

Joan Westfall in story

Attribute	Value
Gender	Female
Father	Jason Westfall
Brother	Brent Westfall
Appearance	Brown eyes

Re^3 (Yang et al. 2022)

Method	ROC-AUC ↑
Random	0.5
GPT-3	0.52
Entailment (Yang et al. 2022)	0.528
Entailment with Dense Passage Retrieval (Yang et al. 2022)	0.610
Attribute Dictionary → Sentence (Yang et al. 2022)	0.684
CoRRPUS (comment)	**0.751**
CoRRPUS (specific)	**0.794**
CoRRPUS (abstract)	**0.704**

Probably because functions like `set_age(self, character, age)` complicate more than they help.

Dong, Y. R., Martin, L. J., & Callison-Burch, C.
“CoRRPUS: Code-based Structured Prompting for Neurosymbolic Story Understanding.” Findings of ACL 2023.

Tricks of the Trade

Instruction-tuned models like GPT-3.5 and Mistral-7B-Instruct like to be given a "role" first (e.g., "You are a helpful writing assistant.")

The more defined the task, the better

- More details
- One thing to do at a time

LLMs are overly confident (like people on the internet)

- To "objectively" have the model evaluate something, you should have another instance judge

Chain-of-thought prompting helps models come up with better answers

They will "Yes and…" your prompt

In-Class Activity

Use GPT-4o (or GPT-4o mini) to generate descriptions of the rooms of the game you made.

Experiment with different types of prompting styles.

https://laramartin.net/interactive-fiction-class/in_class_activities/openai-playground/room-descriptions.html