LLMs and Causality

Tools for Causality Workshop, Autrans 2023

Dr. Robert Osazuwa Ness
Researcher, Microsoft Research
September 12, 2023

Deep dive into LLMs at

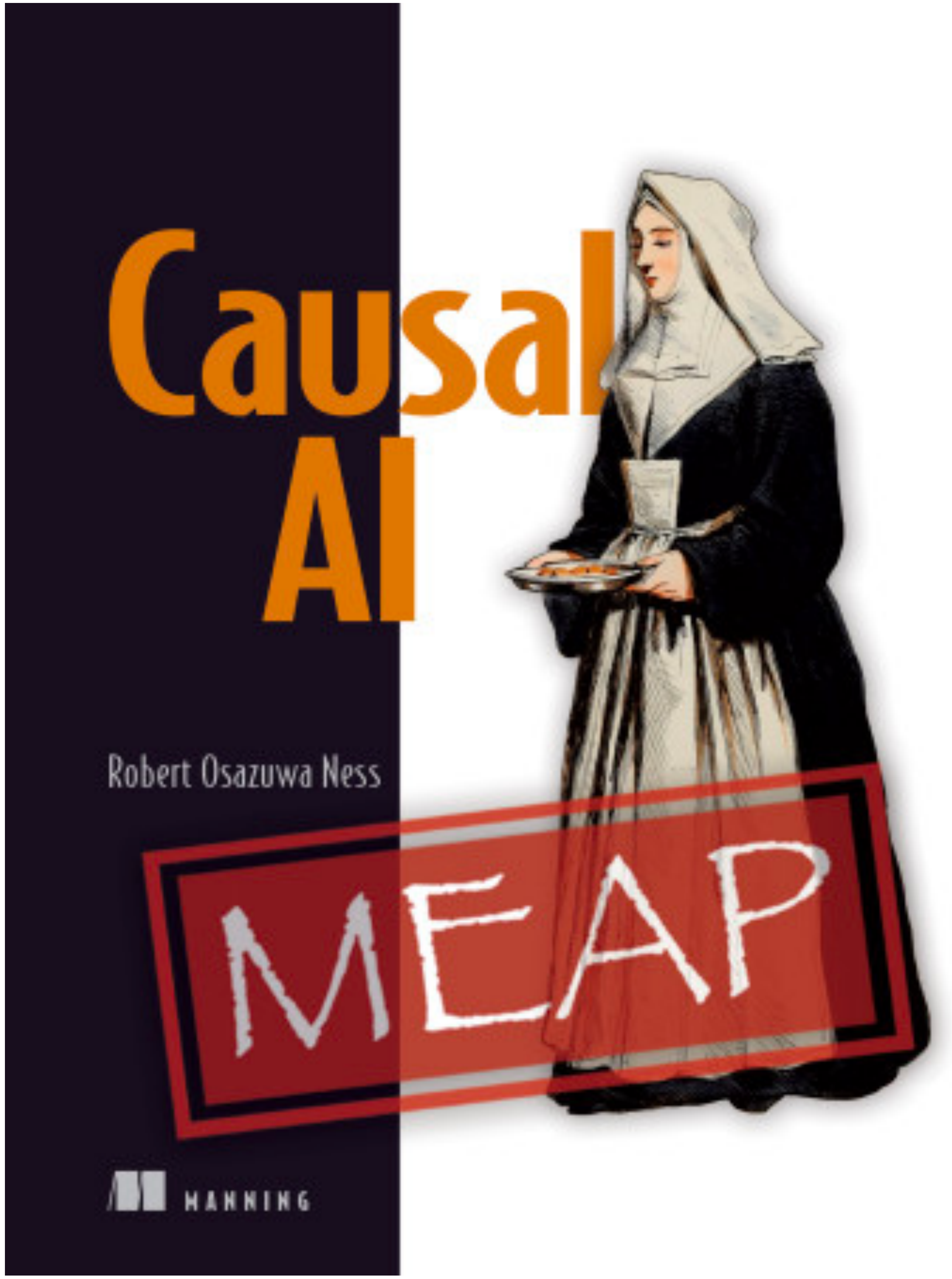
www.altdeep.ai/p/llm

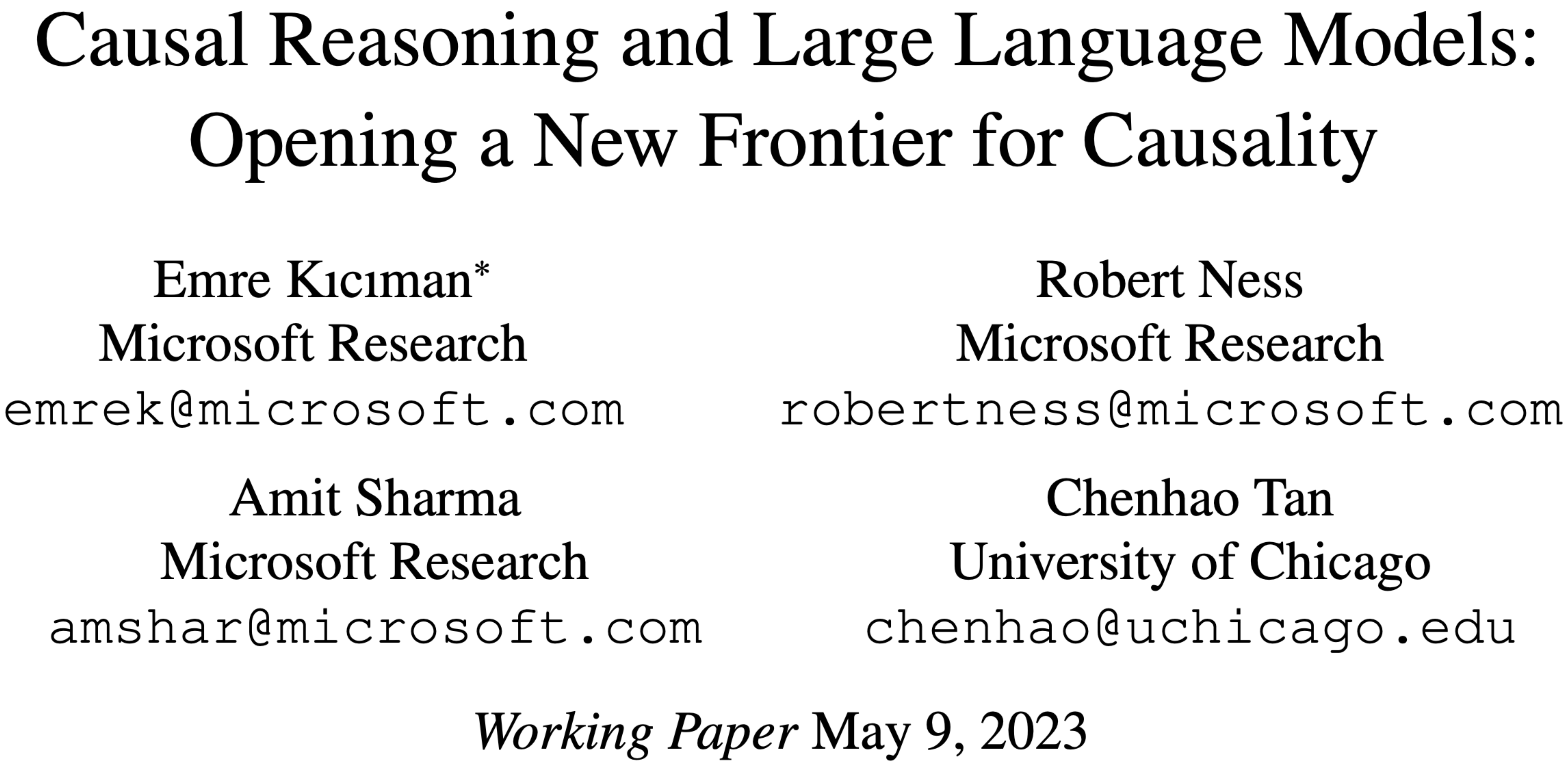
Building Apps with Large Language Models

New cohort starting next week

Book on Causal AI

www.manning.com/books/causal-ai





Causal Reasoning and Large Language Models: Opening a New Frontier for Causality

Emre Kıcıman*
Microsoft Research
`emrek@microsoft.com`

Robert Ness
Microsoft Research
`robertness@microsoft.com`

Amit Sharma
Microsoft Research
`amshar@microsoft.com`

Chenhao Tan
University of Chicago
`chenhao@uchicago.edu`

Working Paper May 9, 2023

https://arxiv.org/pdf/2305.00050.pdf

Ways to use an LLM to augment a causal analysis.

Using an LLM to build a Causal DAG

Building a Causal DAG

- We can use the LLM to augment building a causal DAG for a causal analysis.
- For many, specifying assumptions in the form of a causal DAG is daunting. What if your DAG is wrong?
- Modern LLMs are invaluable as a way of leveraging public expert knowledge to build a DAG.

Building a Causal DAG

User: You are a helpful causal analysis assistant. I'm interested in modeling the relationship between smoking and lung cancer. Provide me a causal DAG that captures this causal relationship.

🤖**ChatGPT**: Certainly! A Directed Acyclic Graph (DAG) is a powerful tool to represent causal relationships. Here's a basic causal DAG to represent the relationship between smoking and lung cancer:

1. **Nodes**:
   - **S**: Smoking behavior (e.g., smoker, non-smoker, or frequency of smoking)
   - **LC**: Lung Cancer (presence or absence)
2. **Directed Edge**:
   - A directed edge from **S** to **LC** represents that smoking behavior can cause or influence the occurrence of lung cancer.
3. **Potential Confounders**:
   - Other nodes can be introduced into the DAG to account for potential confounders. Some potential confounders might be:
     - **G**: Genetics (family history of lung cancer)
     - **A**: Age (older age might be associated with a higher risk)
     - **E**: Environmental exposure (like asbestos, pollution, etc.)
     - **O**: Other health behaviors (like diet, exercise)

   The DAG might look like this with the confounders:

   G –> **LC**
   A –> **LC**
   E –> **LC**
   O –> **LC**
   S –> **LC**

Building a Causal DAG

4. **Potential Mediators**:
   - Mediators are intermediate variables through which a cause exerts its effect on an outcome.
   - For instance, one could hypothesize that smoking might cause lung damage, which in turn causes lung cancer. In this case:
     - **LD**: Lung Damage

       Adding the mediator to the DAG:

       S –> **LD** –> **LC**

(And the confounders would still have their direct relationships with LC as mentioned above.)

Building a Causal DAG

User: Suggest two good candidates for instrumental variables. Be brief in your response.

🤖**ChatGPT**: Certainly. For the relationship between smoking and lung cancer, two potential instrumental variables (IVs) could be:

1. **Cigarette Tax or Price**: If certain regions or periods have higher cigarette prices or taxes, this can influence smoking behavior without directly influencing lung cancer rates except through the effect of reduced smoking. The assumption here is that the only way cigarette prices could affect lung cancer is by influencing smoking behavior.
2. **Anti-Smoking Campaigns or Legislation**: Introduction or intensity of anti-smoking campaigns or legislation in certain regions or times can be used as an instrument. If these campaigns or laws are exogenously determined (not based on current lung cancer rates), they can affect smoking rates without having a direct effect on lung cancer, except through their impact on smoking behavior.

For both IVs, it's crucial to validate the two key assumptions of instrument validity:
1. The instrument is related to the exposure (e.g., cigarette tax affects smoking behavior).
2. The instrument affects the outcome (e.g., lung cancer) only through its effect on the exposure and not through other pathways.

Generating code for DAGs, models, and causal analyses

Generating code for DAGs, models, and causal analyses

- Cutting-edge LLMs are trained on vast corpora of code from a wide variety of programming languages.
- This includes libraries for causal analysis.
- This greatly speeds up causal analyses.
- We can prompt ChatGPT to implement its suggested lung cancer DAG in DoWhy and estimate the causal effect.

Generating code for DAGs, models, and causal analyses

User: Provide me with code that uses the DoWhy library in Python to implement the causal DAG that you suggested above. Include cigarette price as an instrumental variable and provide code for estimating the causal effect.

🤖**ChatGPT**: Certainly! Here’s a basic example using the DoWhy library in Python for implementing the causal DAG for the relationship between smoking and lung cancer, using cigarette price as an instrumental variable.

First, ensure you have installed the DoWhy library:

```
pip install dowhy
```

Then, you can use the following code:

```
import numpy as np
import pandas as pd
import dowhy
from dowhy import CausalModel
from dowhy.datasets import linear_dataset
from dowhy.do_why import CausalModel

# Let's create a sample dataset for illustration purposes
# Normally, you'd import your own dataset: data =
pd.read_csv('your_dataset.csv')
n_points = 1000
data = pd.DataFrame({
    # 0 for non-smoker, 1 for smoker
    "S": np.random.binomial(n=1, p=0.5, size=n_points),
    # 0 for no lung cancer, 1 for lung cancer
    "LC": np.random.binomial(n=1, p=0.5, size=n_points),
    # Example cigarette price
    "Price": np.random.normal(loc=5, scale=1,
    size=n_points),
})
```

```
# Create a causal DAG
model=CausalModel(
        data = data,
        treatment='S',
        outcome='LC',
        # Potential confounders
        common_causes=['G', 'A', 'E', 'O'],
        # Instrumental Variable
        instruments=['Price']
)

# Identify the causal effect
identified_estimand = model.identify_effect(
     proceed_when_unidentifiable=True)

# Estimate the causal effect using Instrumental
Variable method
estimate = model.estimate_effect(
      identified_estimand,
      method_name="iv.instrumental_variable",
      method_params={
           'iv_instrument_name': ‘Price'
      })
# Print the causal effect estimate
print(estimate)
```

Explanations of mechanism

Explanations of mechanism

- DAGs are an intuitive way to represent what causes what
- But DAGs do not tell us anything about the nature of the mechanism
- Structural causal models (SCMs) provide a mathematical way of characterizing the mechanism underlying a causal relationship,
  - but often we don’t have a knowledge of mechanism at level of mathematical detail
  - e.g., what is the equation connecting promotions and engagement in social media?
- LLM leverage common and expert knowledge to provide natural language explanations of causal mechanism

Explanations of mechanism

User: You are a helpful causal explanation generator. Provide a characterization of the causal relationship between ad spending and brand awareness. Be concise enough to fit a single slide in a slide deck.

🤖: Causal Relationship Between Ad Spending and Brand Awareness

- **Monotonicity**: As ad spending increases, brand awareness generally rises and does not diminish.
    - Mechanism: Larger ad spends mean more frequent and widespread exposure of the brand to the target audience.
- **Diminishing Returns**: After a certain point, additional ad spending might not lead to proportionally higher brand awareness.
    - Reason: Market saturation, ad fatigue, or redundancy in audience reach.
- **External Factors**: Ad efficiency isn't solely based on spend.
    - Influences: Ad quality, placement, target audience relevancy, competition, and market conditions can affect the outcome.

Establishing a causal frame

Establishing a causal frame

- Causal frame problem - How do we decide variables are relevant to a causal problem?
  - “Given an enormous amount of stuff, and some task to be done using some of the stuff, what is the relevant stuff for the task?” ~ Clark Glymour
- LLMs are useful at establishing selecting the causal variables relevant to a causal query.
  - i.e., establishing the *causal frame*

Establishing a causal frame

User: You are a helpful causal reasoning assistant. Consider the following story.

Jake is logged into an office network. There was an open shared folder on the server. Jake is transferring files across the network and accidentally clicks on the folder, starting a process that consumes a significant amount of bandwidth. This causes the network to slow down.

How responsible is Jake for the network slowdown?

Establishing a causal frame

🤖**ChatGPT**: Based on the provided story, let's analyze the situation using causal reasoning:

1. Direct Cause: Jake started a process that consumes a significant amount of bandwidth by accidentally clicking on a file.

2. Effect: The network slowed down due to the consumed bandwidth.
   Given these facts, we can establish that Jake's action was the direct cause of the network slowdown.

However, responsibility can be considered from different dimensions:

- Intent: Jake's action was accidental. He did not intend to slow down the network.

- **Prevention Mechanisms**: The story does not specify if there were any warnings or user prompts before starting such a process that consumes significant bandwidth. If there were no warnings, one could argue that the system was not designed to prevent such accidental actions.

Establishing a causal frame

- **Knowledge & Training**: Was Jake trained or informed about the implications of transferring large files or about the particular file that consumes a lot of bandwidth? If not, his knowledge gap contributed to the accident.

- **System Design**: The open shared folder on the server was accessible to Jake. If the file was known to be a bandwidth hog, it's worth asking why such a file was placed in an open shared folder without any warnings or restrictions.

To directly answer the question: Jake is responsible for the network slowdown in the sense that his action was the direct cause. However, the degree of his responsibility might be mitigated if there were systemic design flaws, lack of training, or absence of preventive measures. If other factors, like the ones mentioned above, played a role in the incident, then the responsibility would be shared among Jake and those other factors.

Understanding and contextualizing causal concepts

- It can be challenging to understand formal concepts from causal inference and apply them to real world problems.
- LLMs help explain and contextualize concepts within your problem space.
- Consider an example featuring *necessary causation*.

Understanding and contextualizing causal concepts

User: You are a helpful explainer of causal inference topics. Provide a one paragraph explanation of the concept of necessary causation.

🤖**ChatGPT**: Certainly! Necessary causation refers to a causal relationship where the cause is a necessary condition for the effect to occur. In other words, without the presence of the cause, the effect cannot happen. However, the presence of the cause does not guarantee the effect will occur, as there may be other conditions that also need to be met. Think of it like a key to a lock: the key (cause) is necessary to open the lock (effect), but just having the key doesn't mean the lock will open automatically; you still need to insert and turn it. If the lock remains locked without the key, then the key is a necessary cause for unlocking.

Understanding and contextualizing causal concepts

User: In one paragraph, explain how necessary causation is conceptualized in legal theory.

🤖**ChatGPT**: In legal theory, necessary causation, often termed "but-for causation," is a foundational concept used to establish a direct link between a defendant's actions and the harm suffered by a plaintiff. For an action to be a but-for cause of an outcome, the outcome would not have occurred in the absence of the action. In other words, if one can say "but for the defendant's action, the harm would not have occurred," then necessary causation is established.

Understanding and contextualizing causal concepts

User: Give me an example of necessary causation that involves a Netflix user unsubscribing from Netflix.

🤖**ChatGPT**: A necessary cause for a Netflix user unsubscribing from Netflix might be the exclusive content that is only available on the platform. If a particular show or movie that the user is deeply interested is removed from Netflix while remaining available somewhere else, then the removal of that content becomes a necessary cause for that user's decision to unsubscribe. Without the removal of that content, the user might not have unsubscribed.

Formalization of causal queries

Formalization of causal queries

- For humans, natural language is the medium of our causal reasoning.
- But we need to convert that natural language into formal terms so that we can apply mathematical and algorithmic analysis.
  - This can be a challenge.
  - But fortunately, LLMs can help greatly in this respect.

- Let’s see another example with necessary causation

Formalization of causal queries

User: You are a helpful causal inference assistant. I'm considering a counterfactual inference problem in the context of Netflix. I have the following query: "There is a set of users who unsubscribed after content was removed from the platform. What is the probability a given member of this set of users would not have unsubscribed had the content not been removed?" Write this query as a probabilistic expression with Pearl's counterfactual notation.

Formalization of causal queries

🤖**ChatGPT**: Certainly! Pearl's counterfactual notation offers a concise way to express such queries.

Let's define some variables:

1. X = removal of the content (binary, 1 if removed, 0 if not removed).

2. Y = whether user unsubscribes (binary, 1 if user unsubscribes, 0 if they did not unsubscribe).

Your query is essentially asking for the probability that a user would not have subscribed, had the content not been removed, given that it was removed and they did unsubscribe.

Using Pearl's counterfactual notation, this can be written as:

$$P(Y_{X=0}=0|X=1, Y=1)$$

Formalization of causal queries

Where:

- $Y_{X=0}$ is the counterfactual outcome for Y had X been set to 0 (i.e., the content is not removed).
- $X = 1$ denotes that the content was removed.
- $Y = 1$ denotes that the user unsubscribed.

So, $P(Y_{X=0}=0|X=1, Y=1)$ is the probability that a user would not have unsubscribed had the content not been been removed, given that it was removed and they did unsubscribe.

So the LLM can take us from conceptualization to formalization.

But LLMs also hallucinate.

Beware. LLMs hallucinate.

User: Provide Python code that, for a given pair of treatment and outcome variables, calculates the counterfactual query "probability of necessity."

Beware. LLMs hallucinate.

User: Provide Python code that, for a given pair of treatment and outcome variables, calculates the counterfactual query "probability of necessity."

🤖**ChatGPT**: The Probability of Necessity (PN) is the proportion of cases where the treatment is present, and the outcome is achieved ONLY when the treatment is present. In simpler terms, it indicates the probability that the outcome occurs because of the treatment.

Given that definition, you can calculate PN using the following formula:

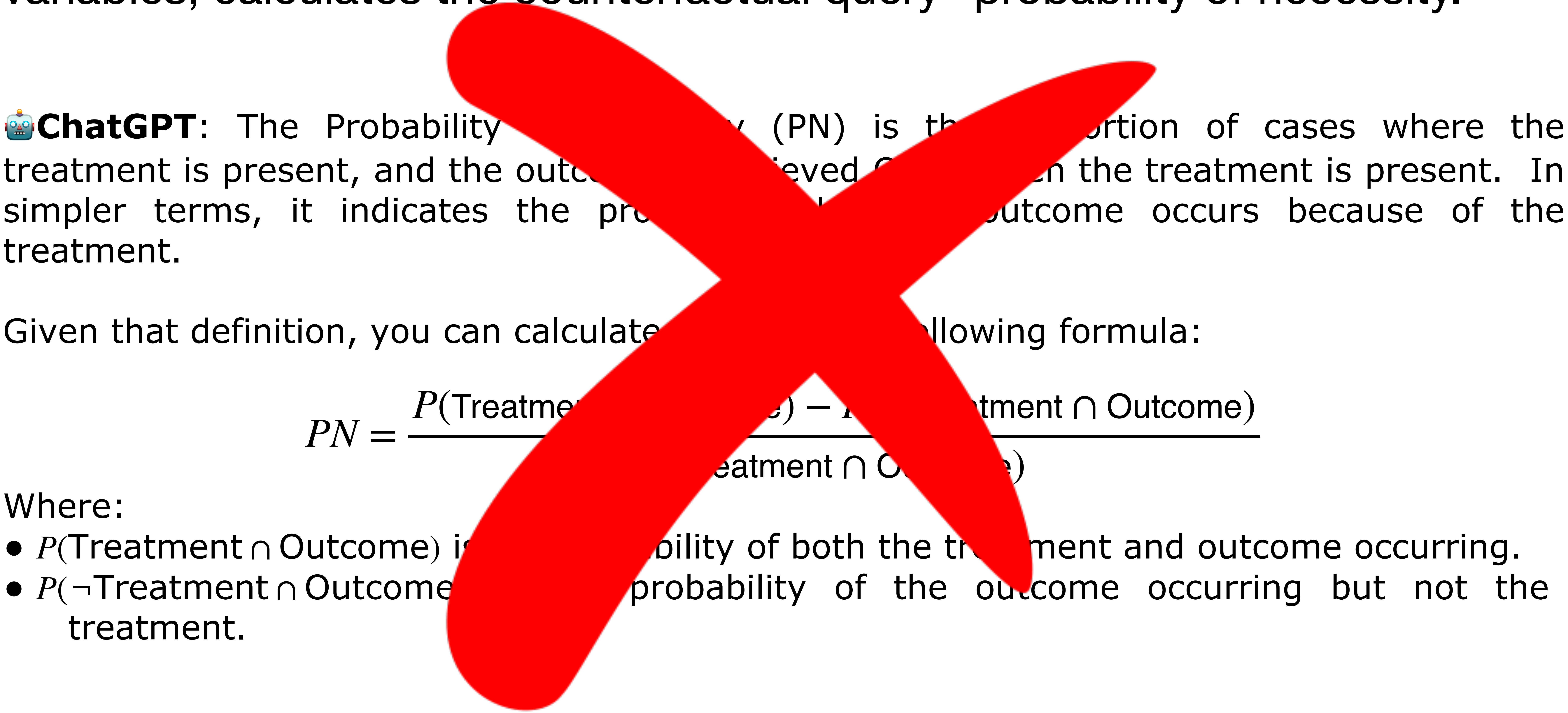
$$PN = \frac{P(\text{Treatment} \cap \text{Outcome}) - P(\neg\text{Treatment} \cap \text{Outcome})}{P(\text{Treatment} \cap \text{Outcome})}$$

Where:

- $P(\text{Treatment} \cap \text{Outcome})$ is the probability of both the treatment and outcome occurring.
- $P(\neg\text{Treatment} \cap \text{Outcome})$ is the probability of the outcome occurring but not the treatment.

Beware. LLMs hallucinate.

User: Provide Python code that, for a given pair of treatment and outcome variables, calculates the counterfactual query "probability of necessity."



🤖**ChatGPT**: The Probability [illegible] (PN) is th[illegible]ortion of cases where the treatment is present, and the outc[illegible]ieved [illegible]n the treatment is present. In simpler terms, it indicates the pr[illegible]utcome occurs because of the treatment.

Given that definition, you can calculate [illegible]llowing formula:

$$PN = \frac{P(\text{Treatme[illegible]e}) - \text{[illegible]tment} \cap \text{Outcome})}{\text{[illegible]eatment} \cap \text{O[illegible]e})}$$

Where:

- $P(\text{Treatment} \cap \text{Outcome})$ i[illegible]bility of both the tr[illegible]ment and outcome occurring.
- $P(\neg\text{Treatment} \cap \text{Outcome}$ [illegible] probability of the ou[illegible]tcome occurring but not the treatment.

Beware. LLMs hallucinate.

The problem is that the response is *close* to right.

- $PN = \frac{P(\text{Treatment} \cap \text{Outcome}) - P(\neg\text{Treatment} \cap \text{Outcome})}{P(\text{Treatment} \cap \text{Outcome})}$ is right only under certain strong assumptions.
  - But these assumptions were not provided in the prompt, nor were they stated explicitly in the generated text.
- LLMs are trained to provide very convincing answers — when they are wrong, they often seem right.
- At present, this means you have to already know the theory pretty well to safely apply it
  - i.e., to be able to detect when the model is hallucination
  - So its not quite at level of automating causal analysis

Forging your own Causal LLM

Run tutorial at bit.ly/causalLLM

Finding structure in the training data

Often training data has structure that is ignored by generative AI

- LLMs and other generative AI are typically trained on “flat” forms of content in their training data
- Documents often have structure:

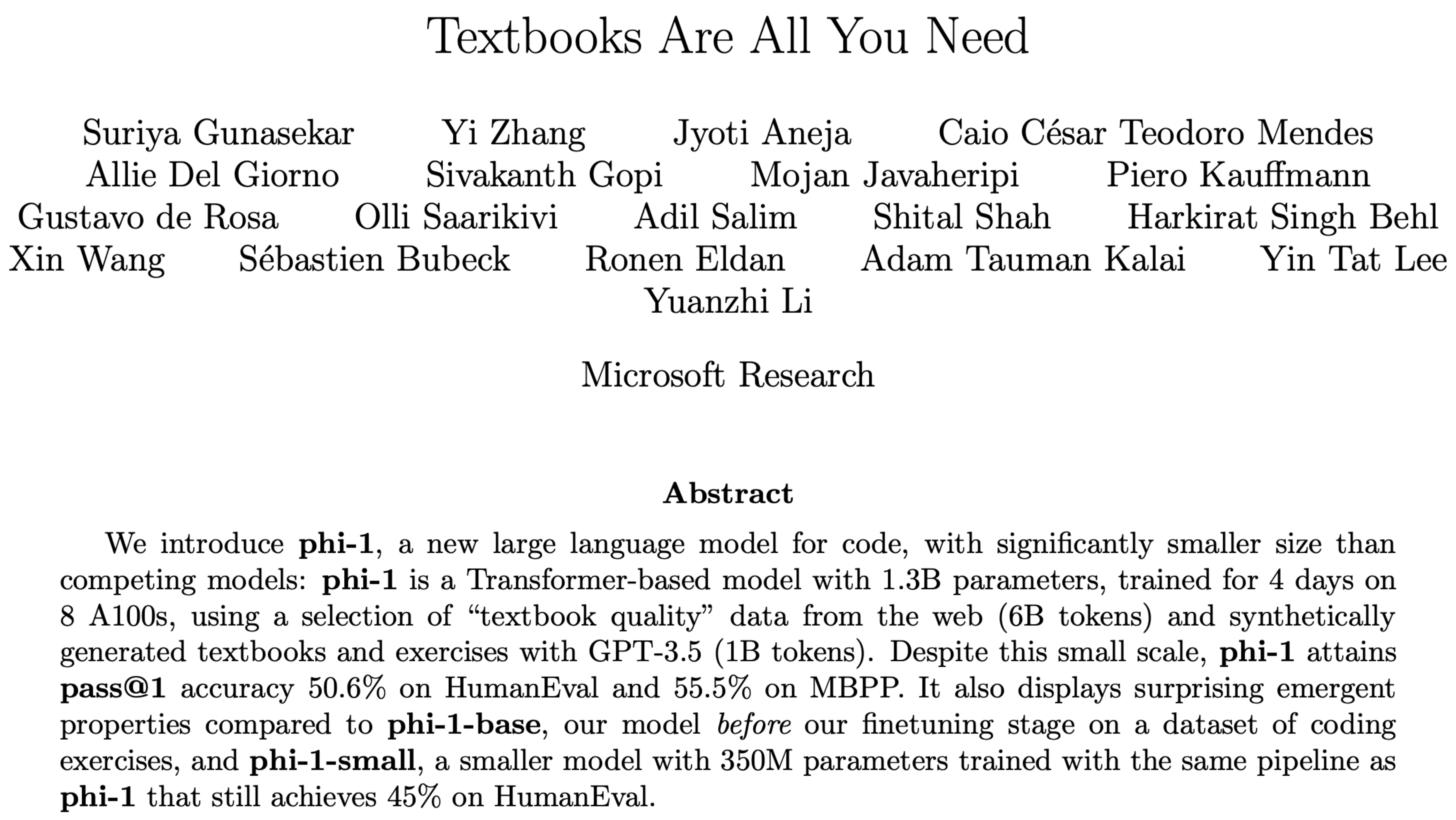
Flat	**Structured**
Newspaper article	Documents and notes that led to the newspaper
Code repository	Commit history of code repository
Illustration	All the layers in the raw design file of the

- Hard to use this structure when you train very large models on a very large corpus

Finding structure in the training data

Often training data has structure that is ignored by generative AI

- As we advance research on how to get better LLM performance with less data and less parameters,…



Textbooks Are All You Need

Suriya Gunasekar Yi Zhang Jyoti Aneja Caio César Teodoro Mendes
Allie Del Giorno Sivakanth Gopi Mojan Javaheripi Piero Kauffmann
Gustavo de Rosa Olli Saarikivi Adil Salim Shital Shah Harkirat Singh Behl
Xin Wang Sébastien Bubeck Ronen Eldan Adam Tauman Kalai Yin Tat Lee
Yuanzhi Li

Microsoft Research

Abstract

We introduce **phi-1**, a new large language model for code, with significantly smaller size than competing models: **phi-1** is a Transformer-based model with 1.3B parameters, trained for 4 days on 8 A100s, using a selection of "textbook quality" data from the web (6B tokens) and synthetically generated textbooks and exercises with GPT-3.5 (1B tokens). Despite this small scale, **phi-1** attains **pass@1** accuracy 50.6% on HumanEval and 55.5% on MBPP. It also displays surprising emergent properties compared to **phi-1-base**, our model *before* our finetuning stage on a dataset of coding exercises, and **phi-1-small**, a smaller model with 350M parameters trained with the same pipeline as **phi-1** that still achieves 45% on HumanEval.

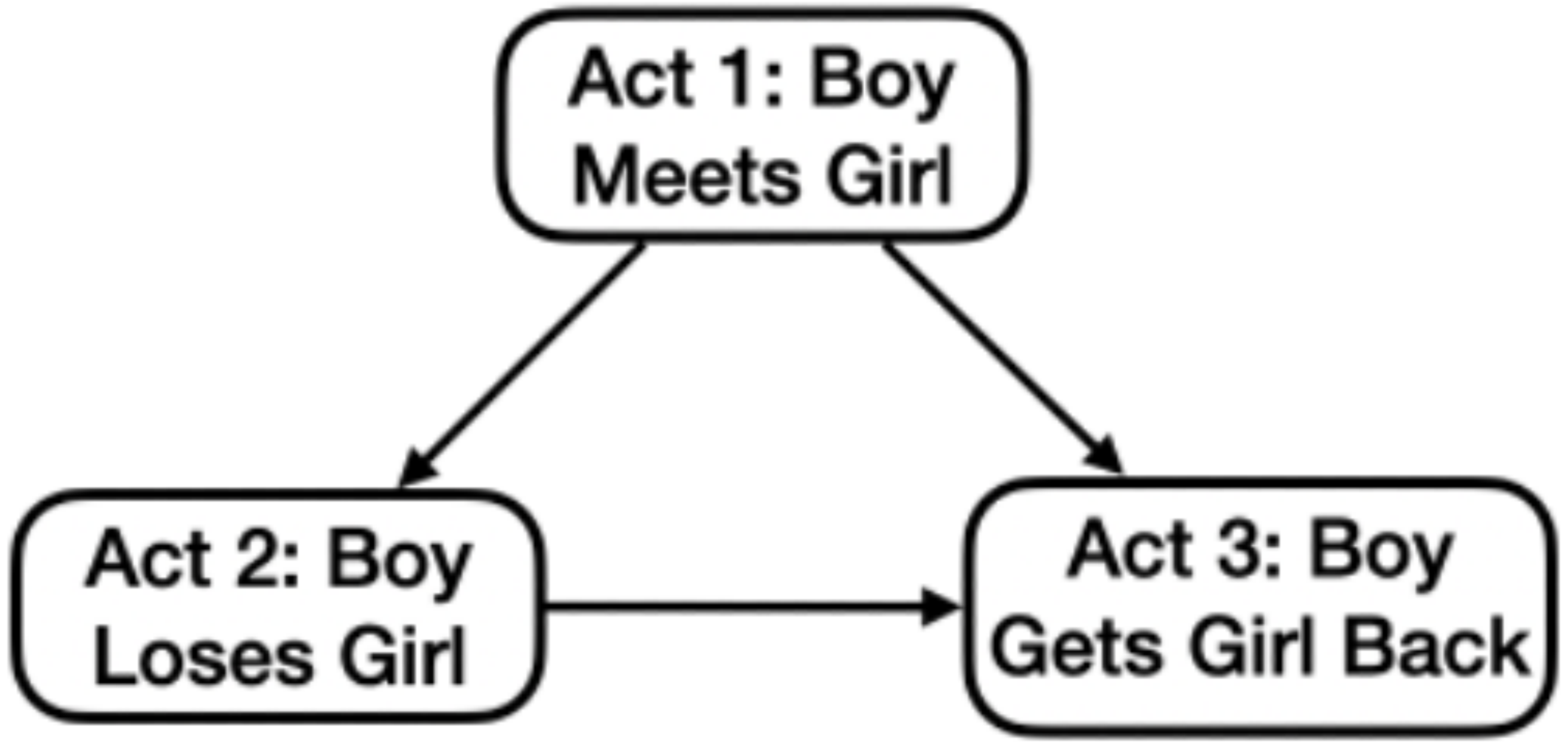
- …there's more opportunity to start to leverage this unused structure.

- Sometimes that structure is causal

Finding structure in the training data

Example: Documents written in script-writing software

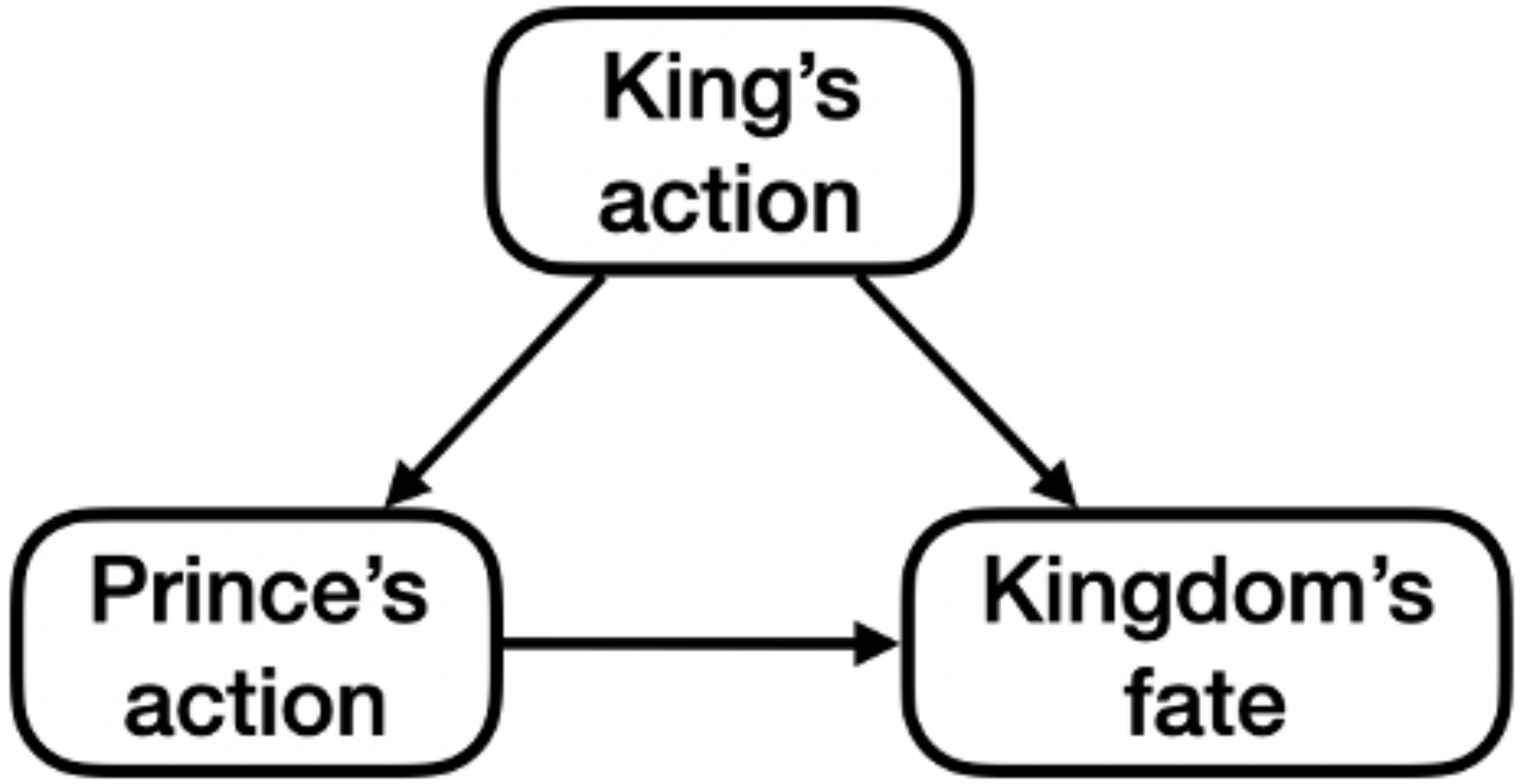
- Script-writing software uses a three-act template
- Events of Act 1 cause events of Act 2, events of Act 1 and 2 cause events of Act 3



Example: King - Prince - Kingdom Model

Example: Imagine a series of scripts about a king, a prince, and a kingdom

- Possible actions of king in Act 1:
  - king declares war
  - king negotiates peace
  - king falls ill
- Possible actions of prince in Act 2
  - prince leads army
  - prince abdicates throne
  - prince marries foreigner
- Possible actions of kingdom in act 3
  - kingdom wins battle;
  - kingdom falls into poverty
  - kingdom experiences prosperity



Example: King - Prince - Kingdom Model

21000 short vignettes

Download CSV Data: https://bit.ly/kingprincekingdom

Example King vignettes

```
King brokers a peace treaty with a rival
kingdom, putting an end to years of bloody
conflict

A wise king successfully negotiates peace
with a rival nation

A wise king successfully negotiates peace
between his kingdom and a long-time enemy
```

Example: King - Prince - Kingdom Model

21000 short vignettes

Download CSV Data: https://bit.ly/kingprincekingdom

Example Prince vignettes

```
however, his son, the Prince, falls in love
and marries a foreigner, causing political
unrest

Prince falls in love with and marries a
foreign princess, forging a strong alliance

but when a new threat emerges, the Prince
leads the army to defend their realm
```

Example: King - Prince - Kingdom Model

21000 short vignettes

Download CSV Data: https://bit.ly/kingprincekingdom

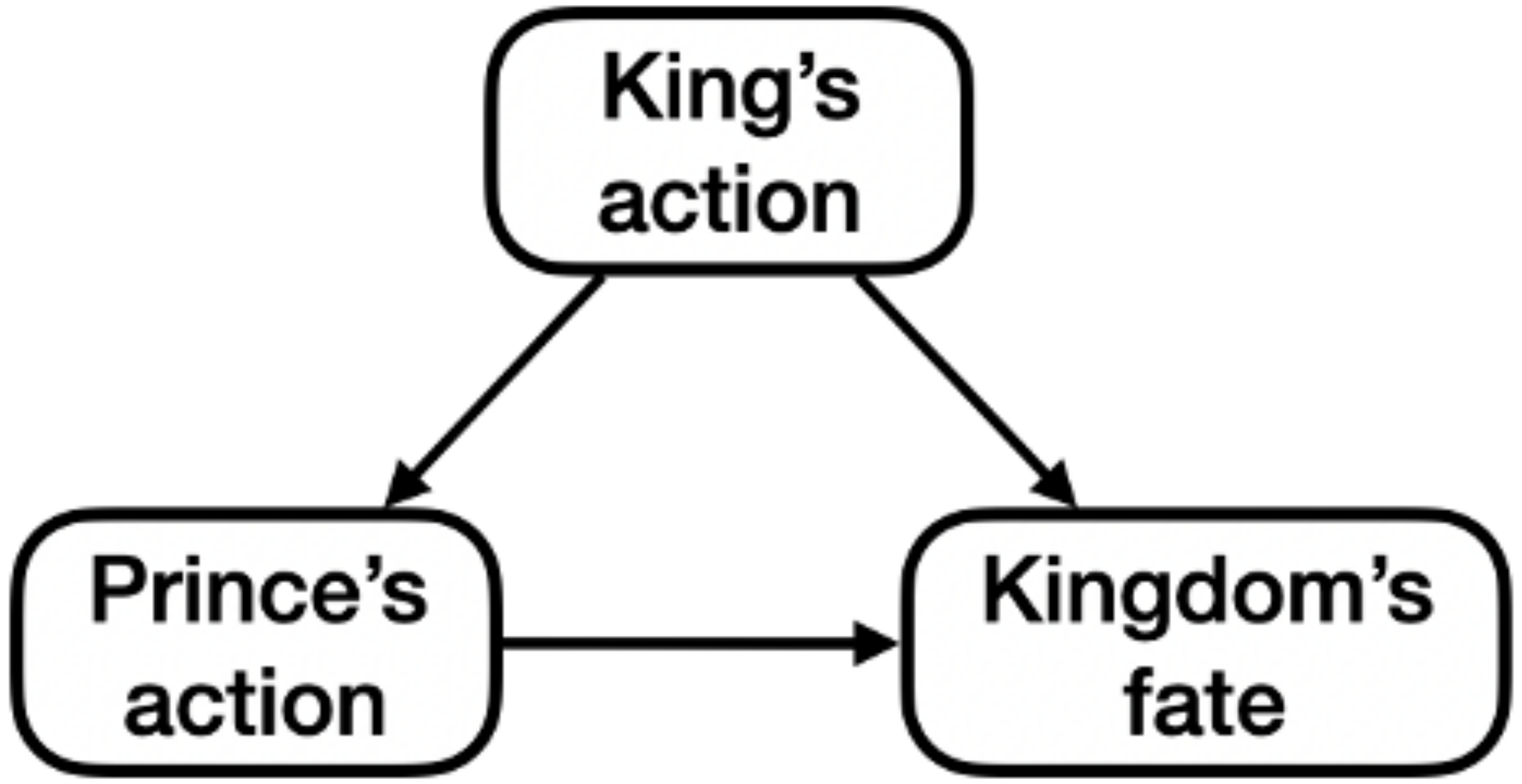
Example Kingdom vignettes

despite efforts, the ongoing war results in both kingdoms falling into poverty.

the alliance strengthens their forces, leading the kingdom to a victorious battle.

however, a series of misfortunes and disastrous decisions plunge their once prosperous kingdom into poverty.

Modeling King, Prince, and Kingdom



Based on the DAG, we need to model:

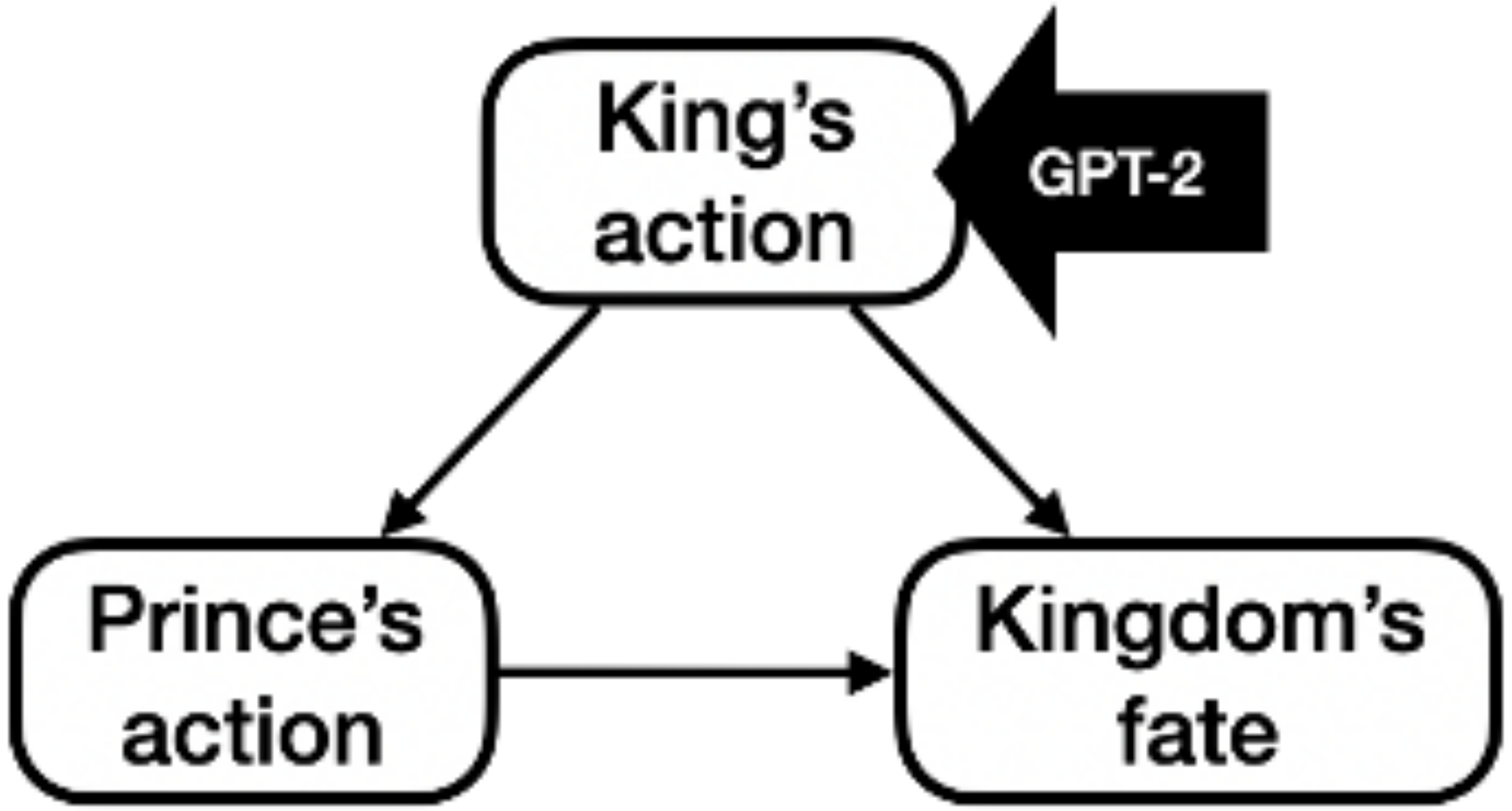
- P(King),
- P(Prince|King=k),
- P(Kingdom|King=k, Prince=p)

Run tutorial at bit.ly/causalLLM

Example: King - Prince - Kingdom Model

Modeling P(King) with a GPT-2 model fine-tuned to the King vignettes to model

```
king_model = AutoModelForCausalLM.from_pretrained("osazuwa/causalLLM-king")
```

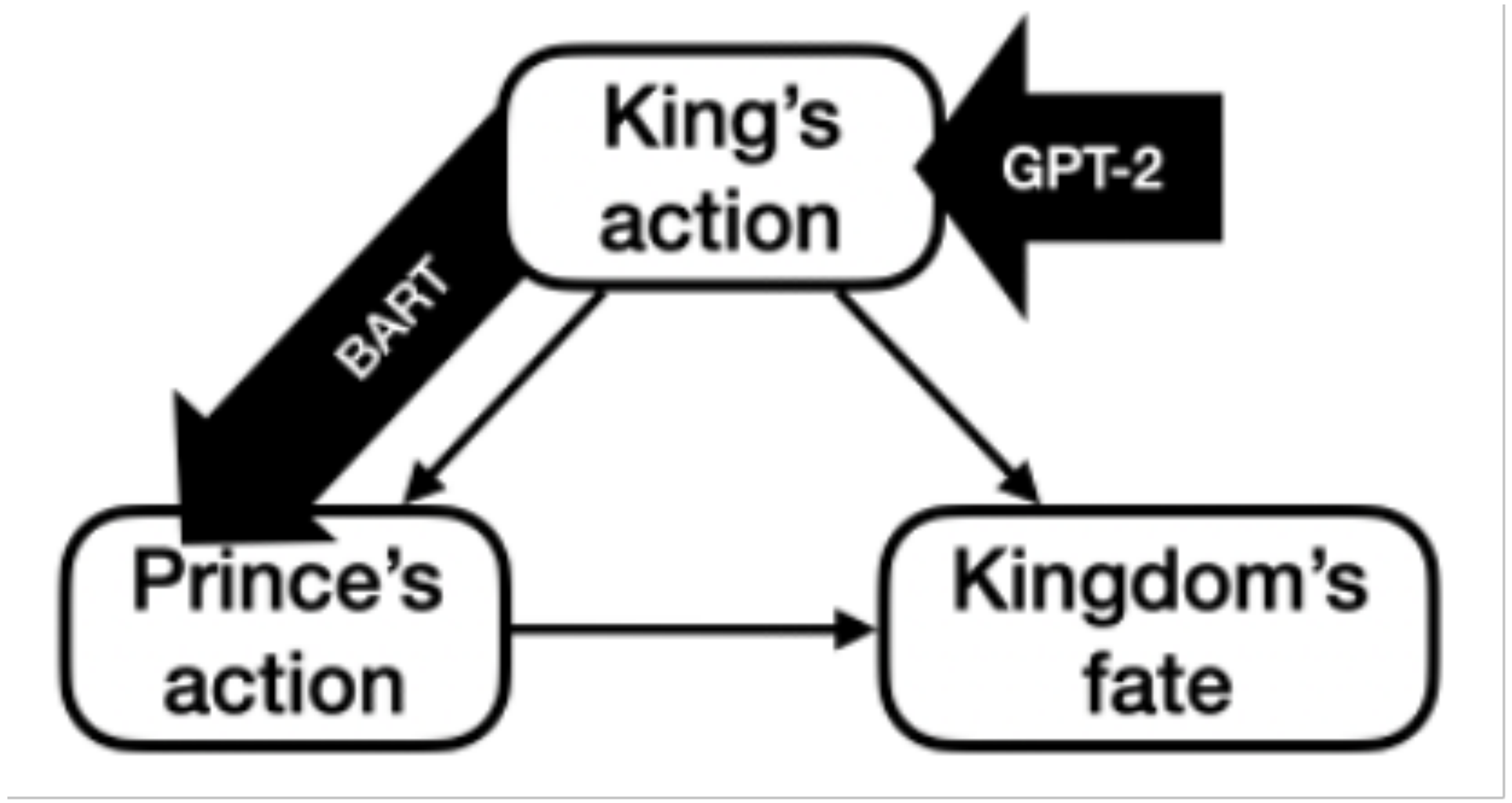


Run tutorial at bit.ly/causalLLM

Example: King - Prince - Kingdom Model

Modeling P(R|K=k) with a fined-tuned BART (Seq2Seq) model that maps King's action vignettes to Prince's action vignettes

```
prince_model = AutoModelForSeq2SeqLM.from_pretrained("osazuwa/causalLLM-prince")
```

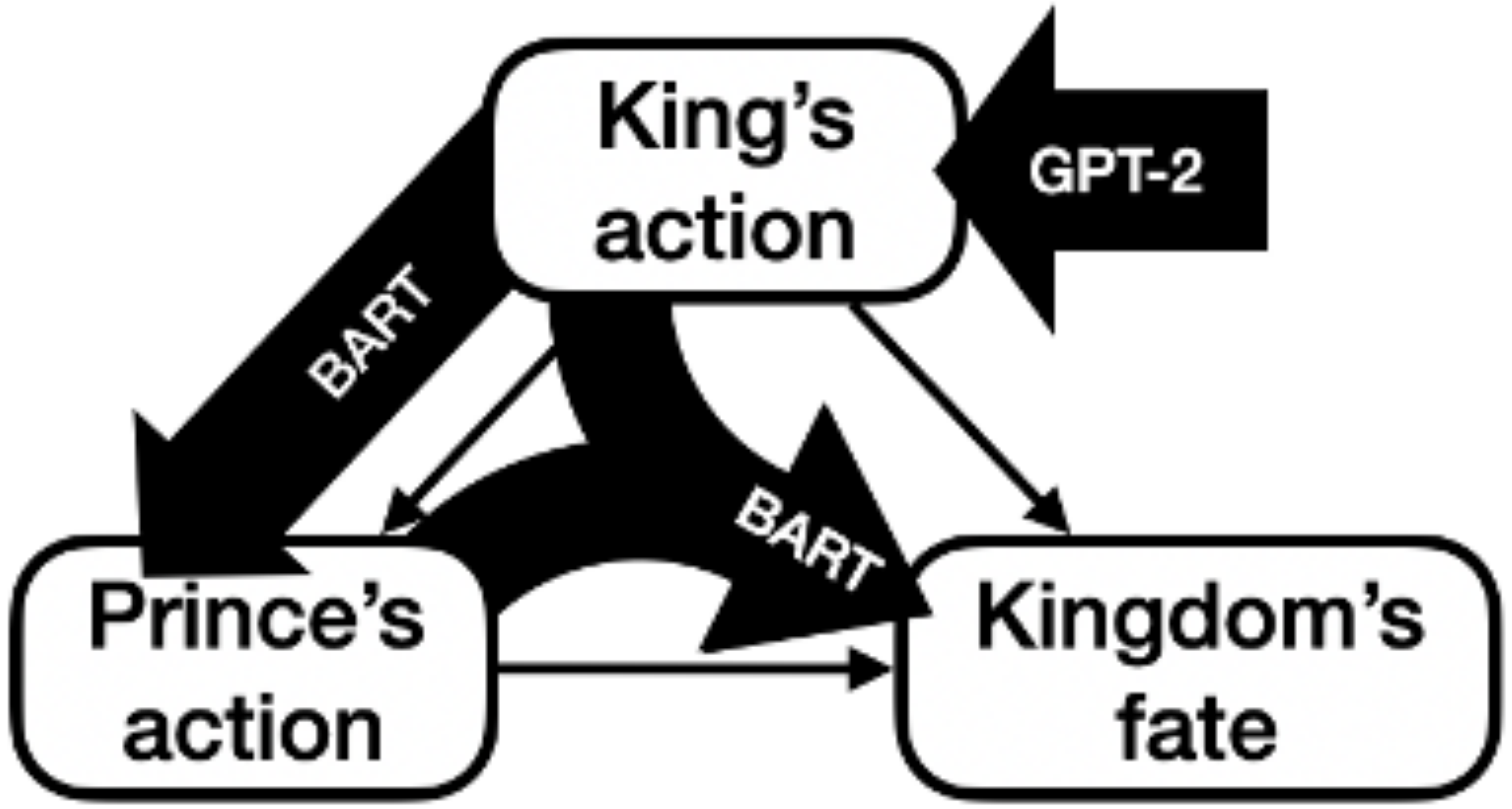


Run tutorial at bit.ly/causalLLM

Example: King - Prince - Kingdom Model

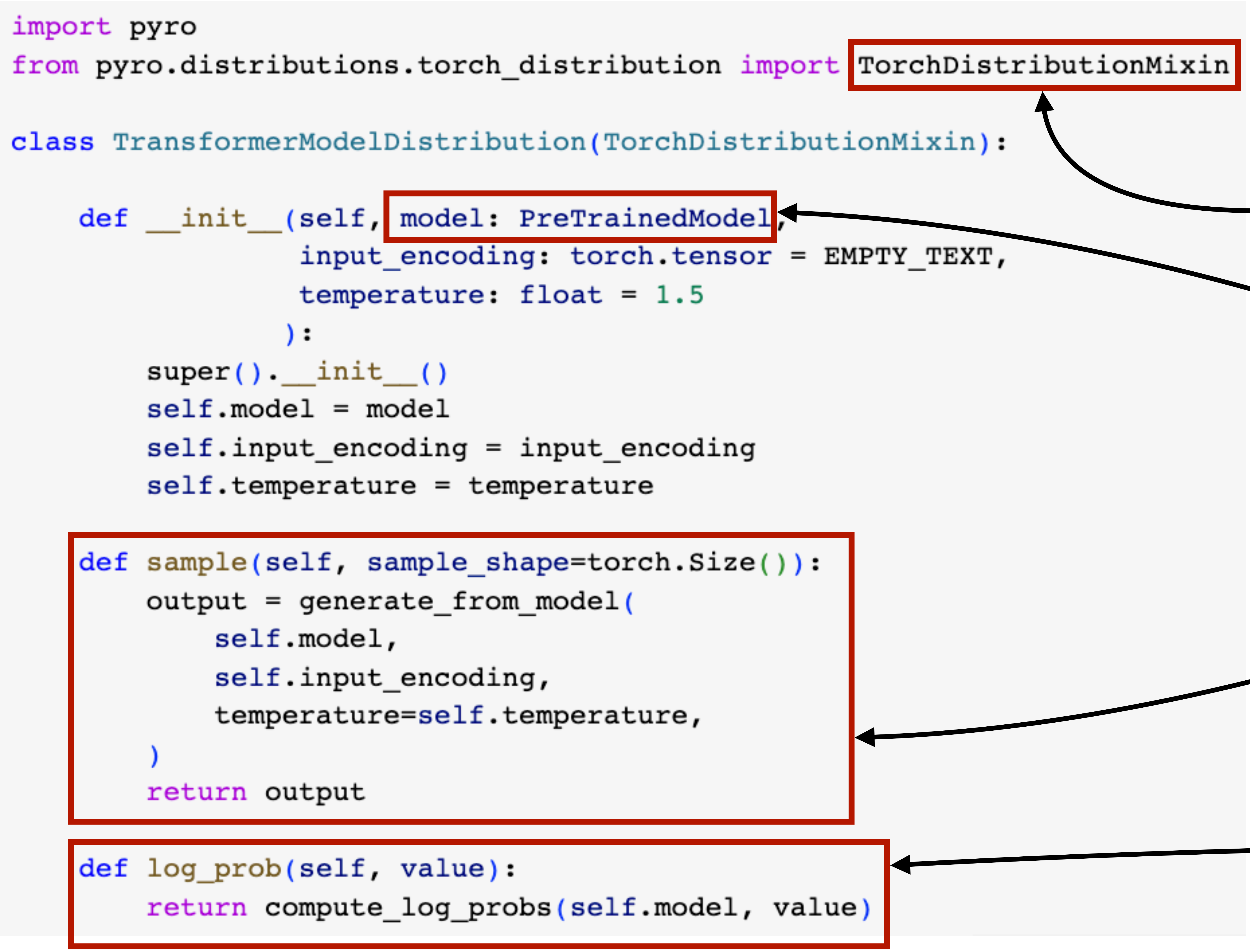
Modeling P(R|K=k) with a fined-tuned BART (Seq2Seq) model that maps King's action vignettes to Prince's action vignettes

```
kingdom_model = AutoModelForSeq2SeqLM.from_pretrained("osazuwa/causalLLM-kingdom")
```



Run tutorial at bit.ly/causalLLM

Use Torch to model each probability distribution



```python
import pyro
from pyro.distributions.torch_distribution import TorchDistributionMixin

class TransformerModelDistribution(TorchDistributionMixin):

    def __init__(self, model: PreTrainedModel,
                 input_encoding: torch.tensor = EMPTY_TEXT,
                 temperature: float = 1.5
                ):
        super().__init__()
        self.model = model
        self.input_encoding = input_encoding
        self.temperature = temperature

    def sample(self, sample_shape=torch.Size()):
        output = generate_from_model(
            self.model,
            self.input_encoding,
            temperature=self.temperature,
        )
        return output

    def log_prob(self, value):
        return compute_log_probs(self.model, value)
```

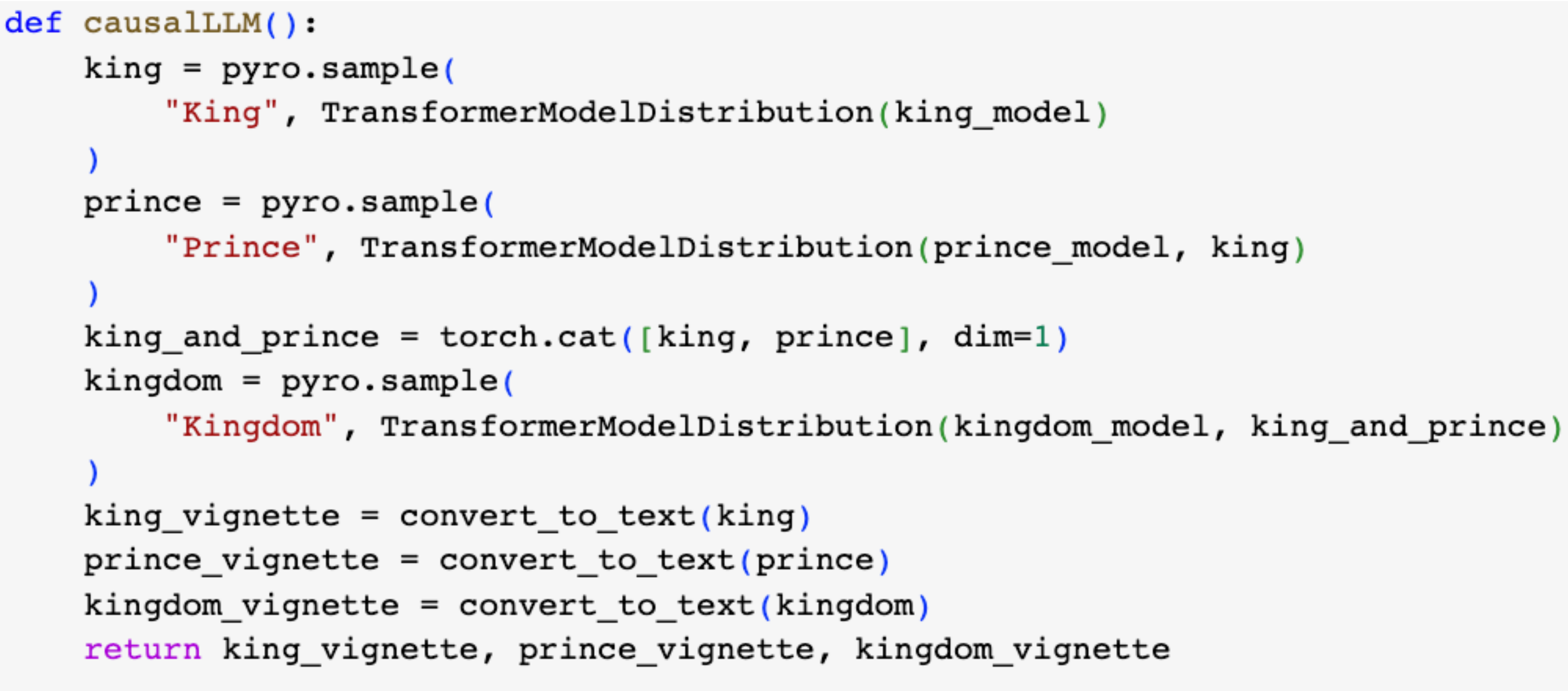
Use Pyro libraries TorchDistributionMixin to create torch distributions that can model P(K), P(R|K=k) and P(D|R=r, K=k)

We'll create a custom class that takes in a pre-trained LLM

Just need to generate text…

… and the ability to calculate log probabilities of generated text

Write a probabilistic model representing the causal DAG

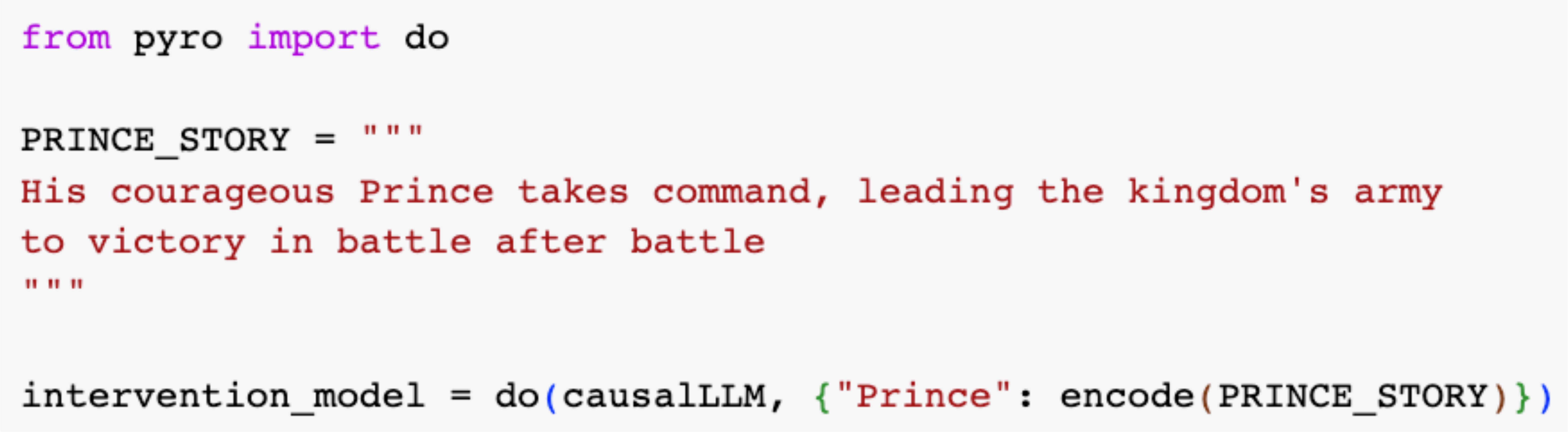


```python
def causalLLM():
    king = pyro.sample(
        "King", TransformerModelDistribution(king_model)
    )
    prince = pyro.sample(
        "Prince", TransformerModelDistribution(prince_model, king)
    )
    king_and_prince = torch.cat([king, prince], dim=1)
    kingdom = pyro.sample(
        "Kingdom", TransformerModelDistribution(kingdom_model, king_and_prince)
    )
    king_vignette = convert_to_text(king)
    prince_vignette = convert_to_text(prince)
    kingdom_vignette = convert_to_text(kingdom)
    return king_vignette, prince_vignette, kingdom_vignette
```

Run tutorial at bit.ly/causalLLM

An LLM where one can apply a do-operator

Can intervene on this model like any causal graphical model

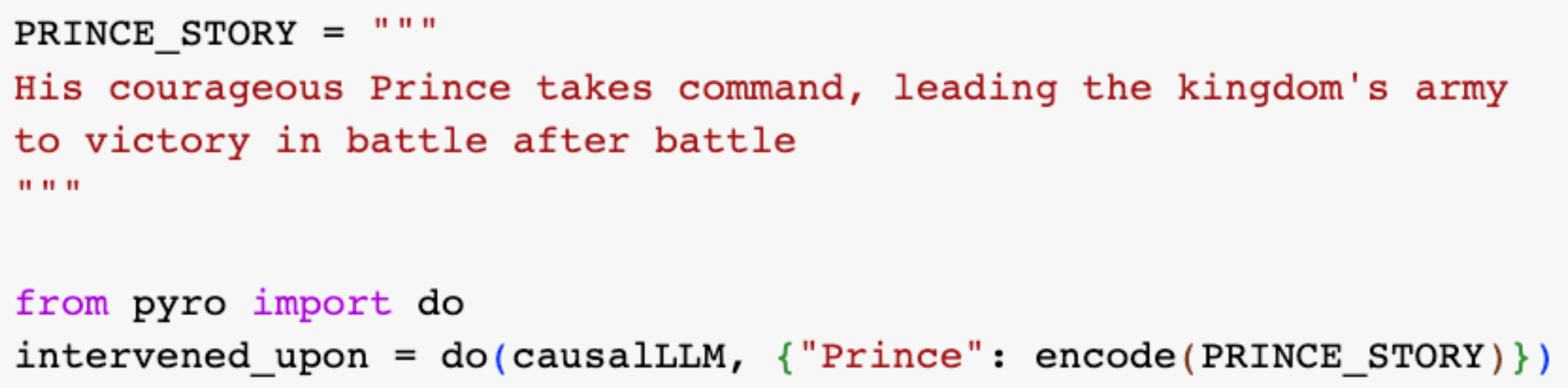


```
from pyro import do

PRINCE_STORY = """
His courageous Prince takes command, leading the kingdom's army
to victory in battle after battle
"""

intervention_model = do(causalLLM, {"Prince": encode(PRINCE_STORY)})
```

Run tutorial at bit.ly/causalLLM

An LLM where one can apply a do-operator

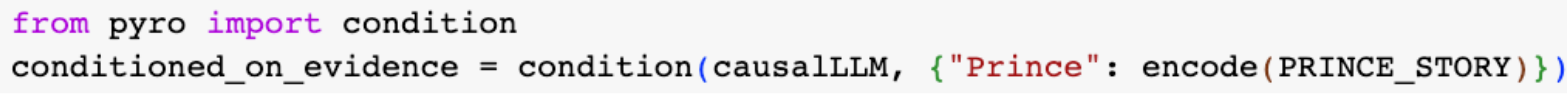
Can intervene on this model like any causal graphical model



```
PRINCE_STORY = """
His courageous Prince takes command, leading the kingdom's army
to victory in battle after battle
"""

from pyro import do
intervened_upon = do(causalLLM, {"Prince": encode(PRINCE_STORY)})
```

Can compare intervention distribution P(Kingdom|do(Prince = prince_story))

to conditional distribution P(Kingdom|Prince=prince_story))



```
from pyro import condition
conditioned_on_evidence = condition(causalLLM, {"Prince": encode(PRINCE_STORY)})
```

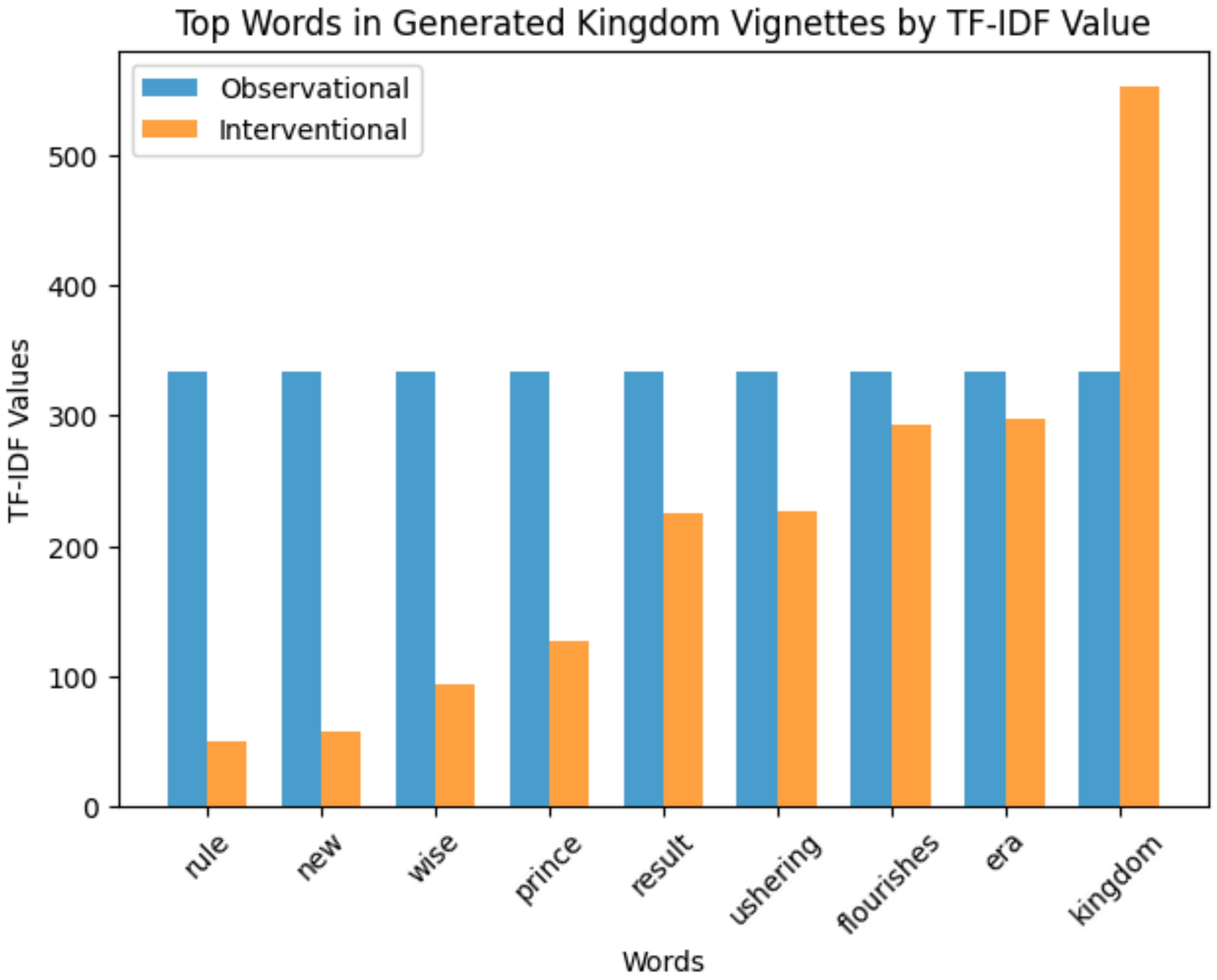
Run tutorial at bit.ly/causalLLM

Comparing interventional and observational distributions

Can compare intervention distribution P(Kingdom|do(Prince = prince_story))

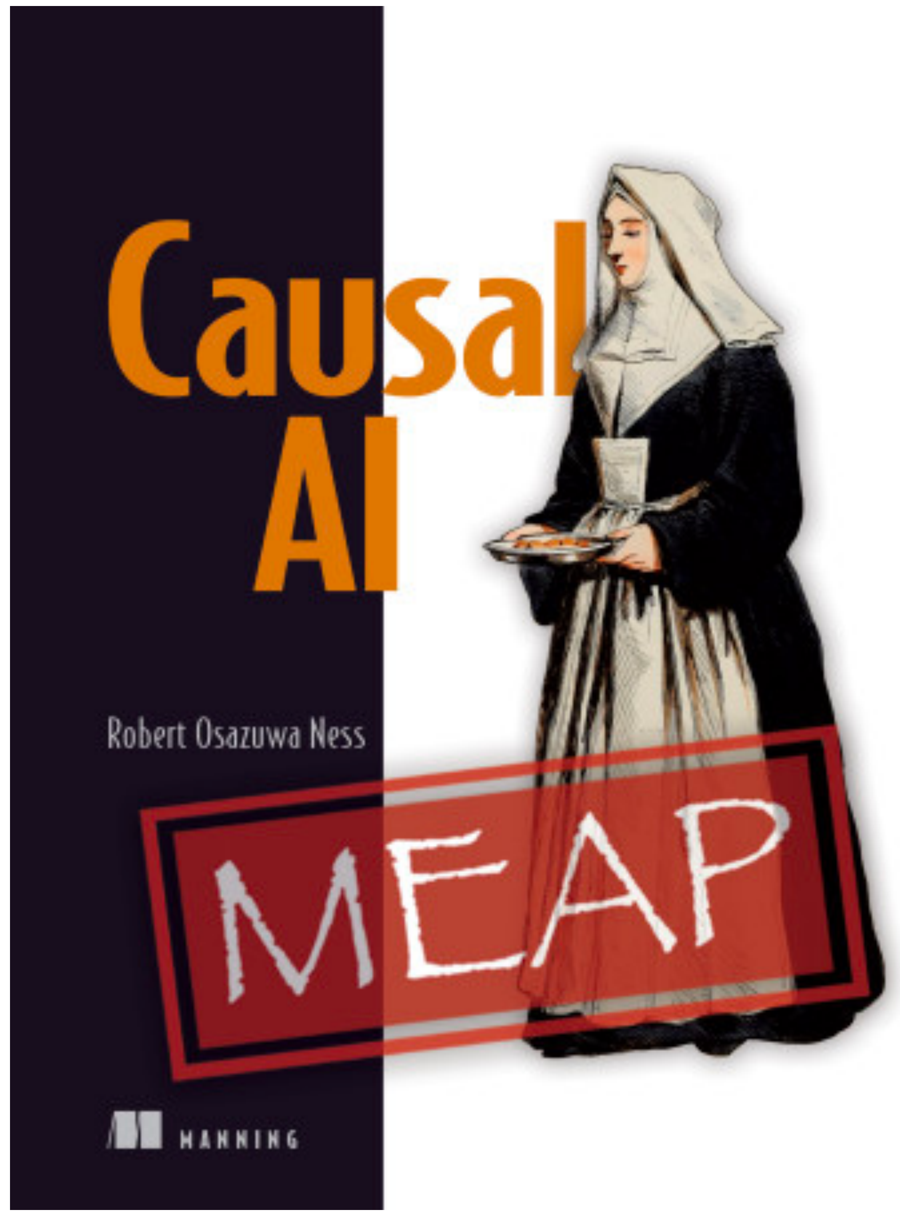
to conditional distribution P(Kingdom|Prince=prince_story))

by generating from both distributions and looking at top terms by TF/IDF



Questions

- Tutorial: bit.ly/causalLLM
- Causal AI Book: www.manning.com/books/causal-ai
- Altdeep.ai LLM course: www.altdeep.ai/p/llm



- Kıcıman, Emre, Robert Ness, Amit Sharma, and Chenhao Tan. "Causal reasoning and large language models: Opening a new frontier for causality." arXiv preprint arXiv:2305.00050 (2023).

  https://arxiv.org/abs/2305.00050