

Ontologies + Recap

10/31/2024

CMSC 491/691 - INTERACTIVE FICTION AND TEXT GENERATION
DR. LARA J. MARTIN

ONTOLOGY SLIDES FROM DR. SUSAN BROWN

Learning Objectives

- Tie together ontologies and knowledge graphs
- Revisit concepts throughout the semester so far
- Pull together concepts and themes you've seen in class
- VOTE

Semantic representations and predicate logic

- *Franco likes Frasca.*

- First order logic:

$$\exists e \text{Liking}(e) \wedge \text{Liker}(e, \text{Franco}) \wedge \text{Liked}(e, \text{Frasca})$$

- VerbNet:

The lion tamer jumped the lion through the hoop.

has_location(e1, Theme, Initial_Location)

do(e2, Agent)

motion(e3, Theme, Trajectory)

has_location(e4, Theme, Destination)

cause(e2, e3)

Semantics

- Let's start with the basics of what we might want to say about some world.
 - There are entities in this world.
 - We'd like to assert properties of these entities.
 - And we'd like to assert relations among them.
- Let's call a scheme that can capture these things *a model*
- And let's claim that we can use basic *set theory* to represent such models.
- We can do this with *an ontology*.

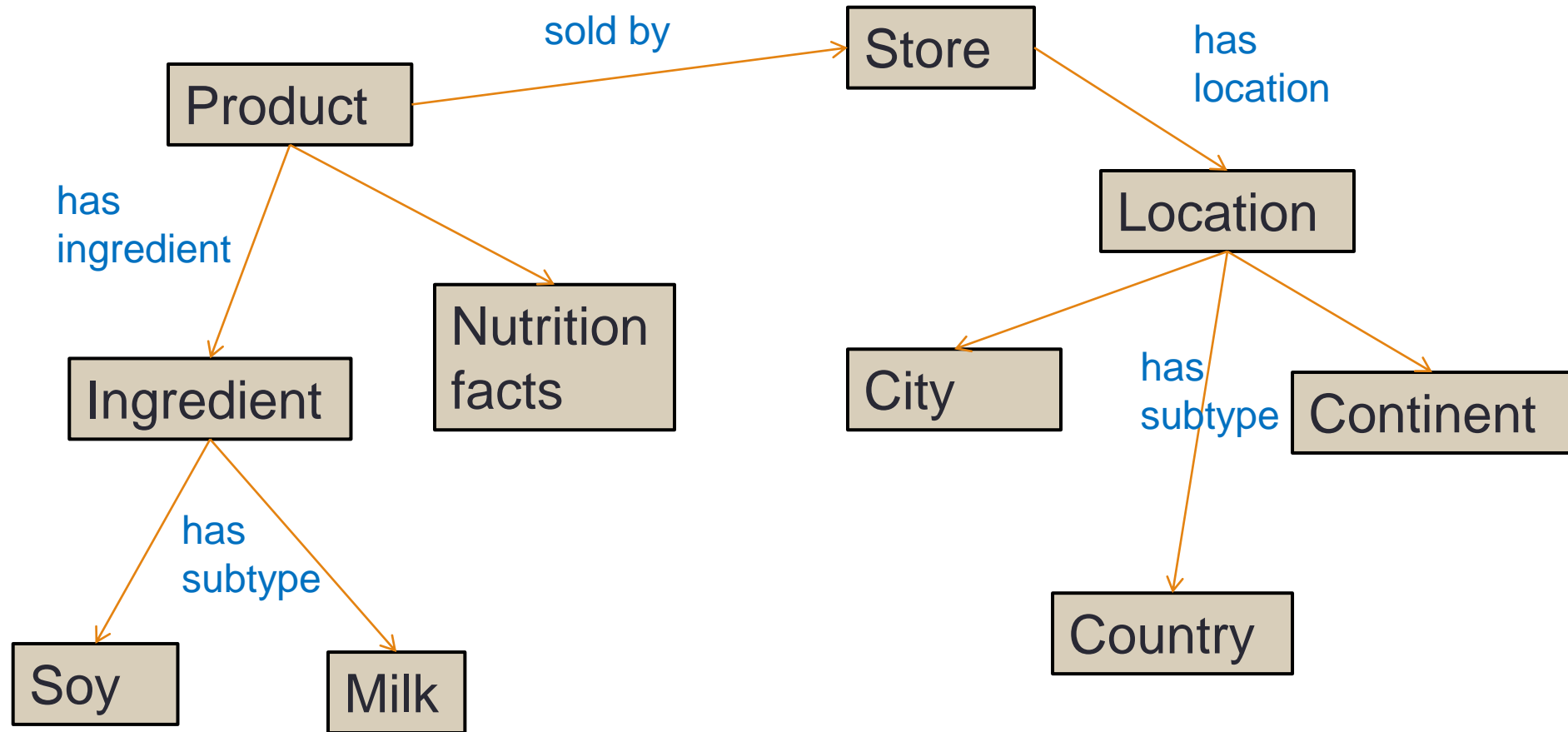
From vocabulary to ontology

- Vocabulary
 - Fixed set of terms
- Taxonomy
 - Fixed set of terms with subset relations between terms
- Ontology
 - Fixed set of terms with structured relationships between terms, generalization, specialization of terms
- Logic-based ontology
 - Ontology that is written in a formal language that is underpinned by a logic, giving it a precisely specified semantics, and computable relationships between terms

What is an ontology

- Describes a domain
 - concepts
 - properties and attributes of those concepts
 - constraints on properties and attributes
 - individuals
- Defines
 - a common vocabulary
 - a shared understanding
- Can be used with reasoning agents
 - to infer new facts from existing definitions

Imagine a mind map for the domain



Ontology basics (using OWL)

Axioms

Basic **statements** in an ontology.
An ontology is a set of axioms

Entities

Used to refer to basic **things in the domain** of interest.

Class Expressions

Combinations of entities that form more **complex descriptions out of simpler ones.**

Axioms specify the relationships between entities and class expressions

OWL Axioms

Some examples...

Cat **SubClassOf** Animal

SubClassOf
Cats are Animals

Cat **DisjointWith** Dog

DisjointClasses
Cats are not Dogs

Tibbs **Type** Cat

ClassAssertion
Tibbs is a Cat

Betty hasPet Tibbs

PropertyAssertion
Betty has Tibbs as a pet

hasPet **Domain** Person

Domain
Anything that has a pet is
Person

Class expressions

Some examples...

Cat **or** Dog

The class of individuals that instances of Cat or Dog (or both!)

Person **and** PetOwner

The class of individuals that are both instances of Person and PetOwner

hasPet **some** Cat

The class of individuals that have at least one hasPet relationship to an individual that is an instance of Cat

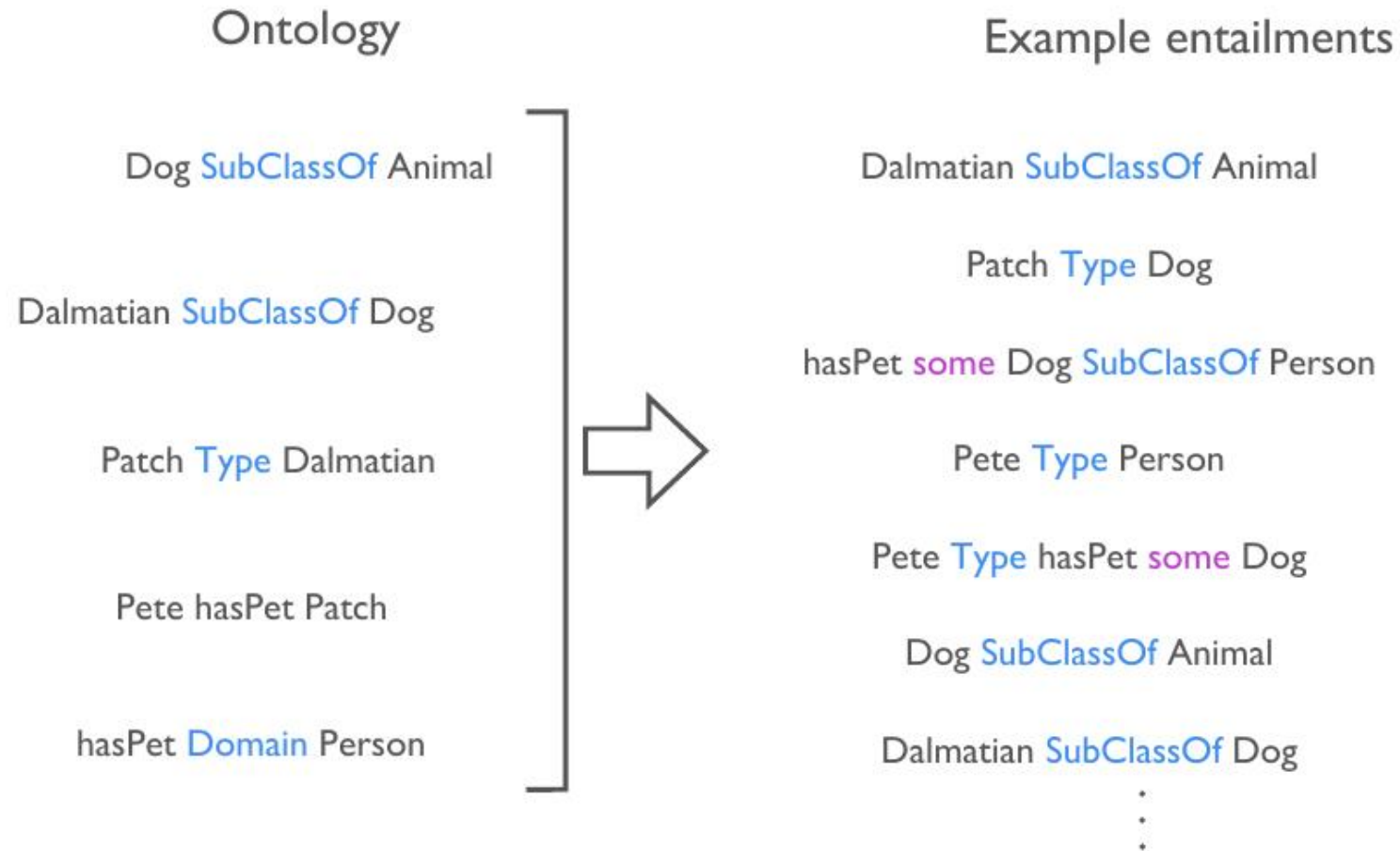
Person **and** hasPet **some** Cat

The class of individuals that are both instances of Person and hasPet some Cat

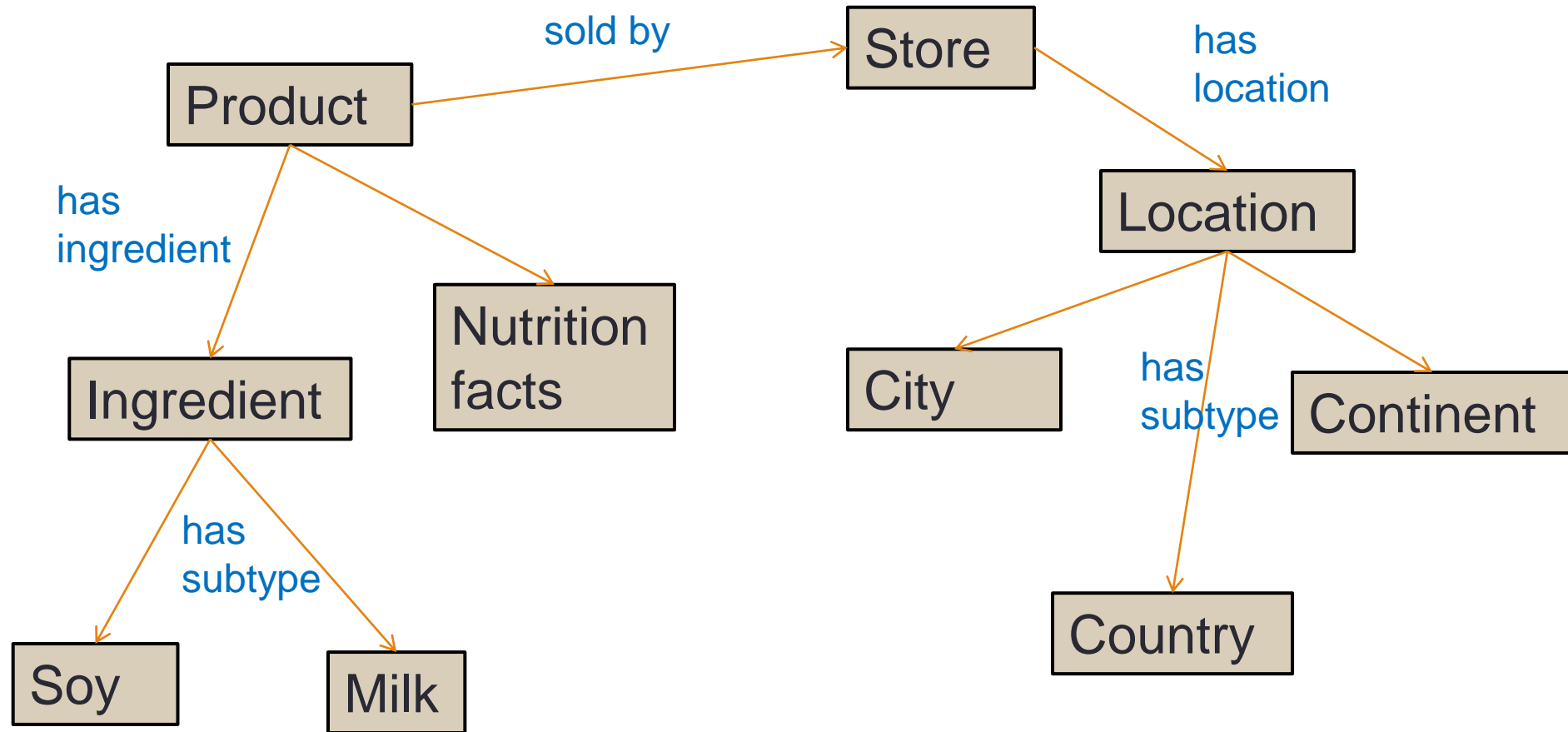
Person **and not** (hasPet **some** (Cat **or** Dog))

The class of individuals that are instances of Person but not instances of the class of individuals that have at least one hasPet relationship to and individual that is an instance of the class Cat or Dog

Entailment



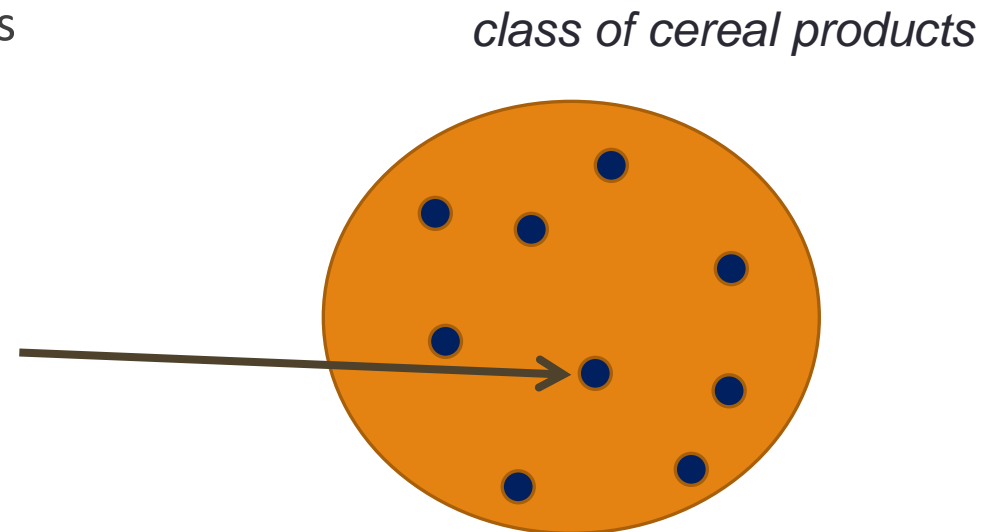
Imagine a mind map for the domain



Defining classes

- A class is a concept in the domain
 - a class of products
 - a class of ingredients
 - a class of dairy products
- A class is a set of elements with similar properties
- Instances of classes
 - a box of cereal that you are buying

*box of cereal
you just bought*

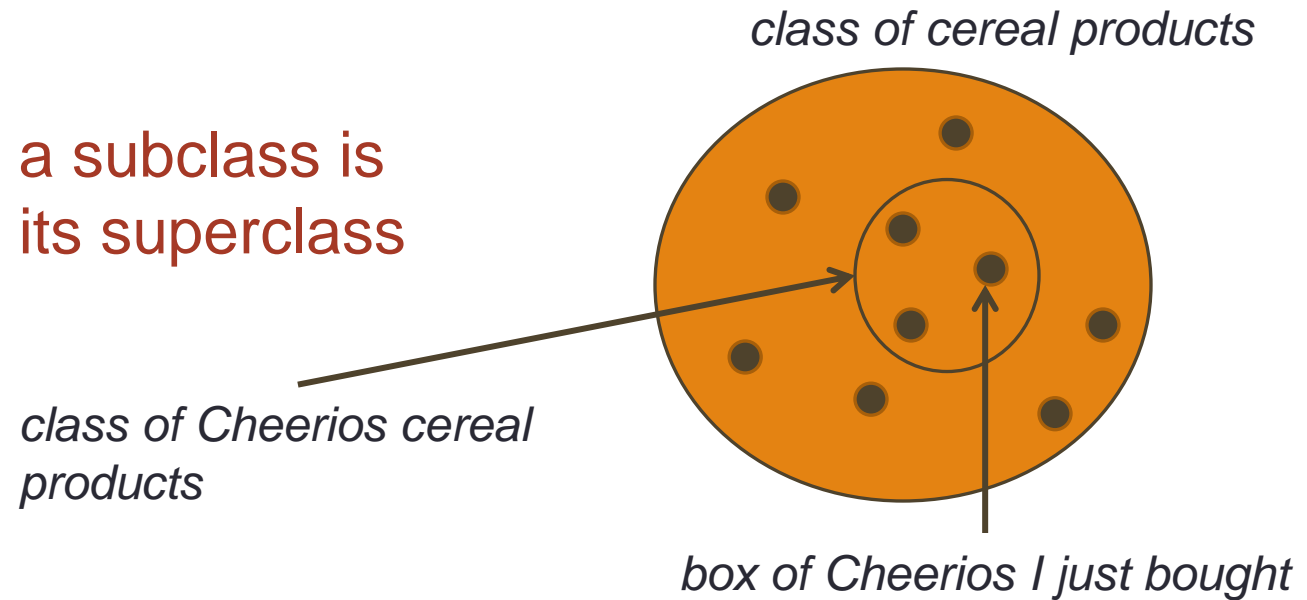


Class inheritance

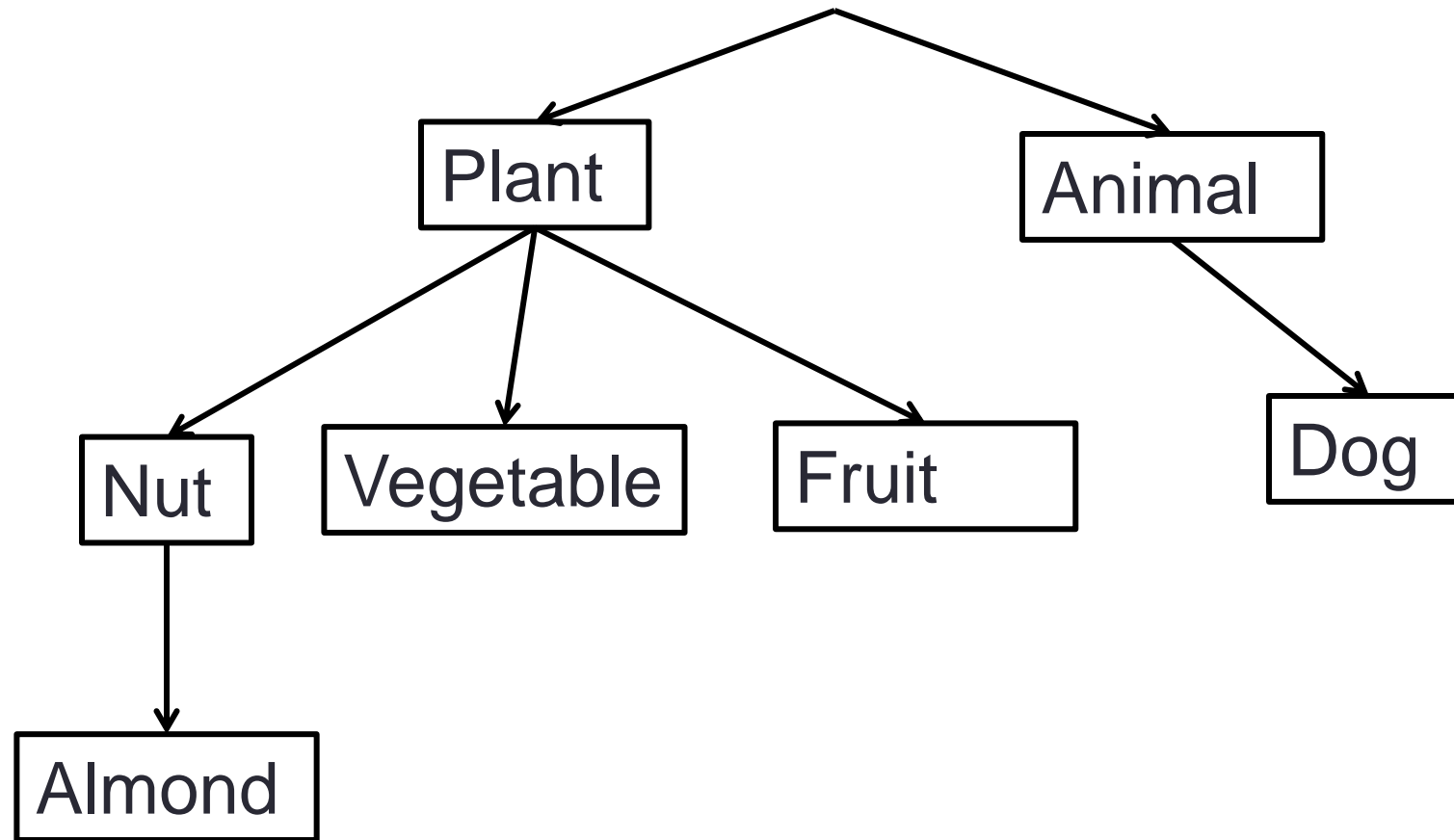
Classes usually constitute a taxonomic hierarchy (a subclass-superclass hierarchy)

an IS-A hierarchy:

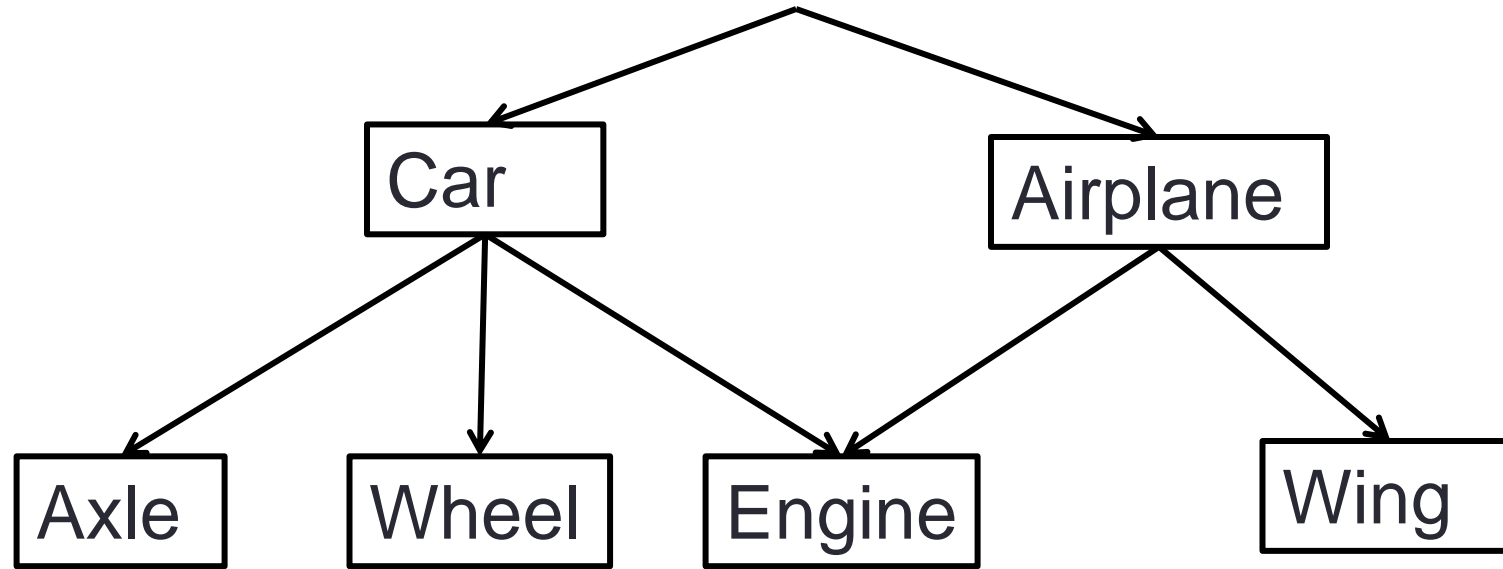
an instance of a subclass is
an instance of its superclass



- If you think of a class as a set, a subclass is a subset



**Subclass-Superclass
relations?**



**Subclass-Superclass
relations?**

Defining properties

Products *have a price*



price

Products *are produced by* a manufacturer



produced by

Products *have an expiration date*



has expiration
date

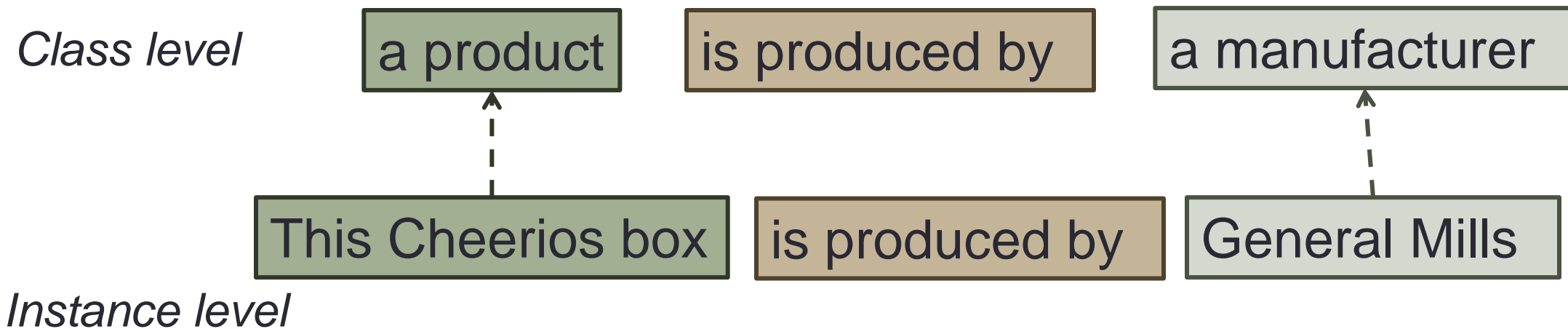
Products *have ingredients*



has ingredient

Properties describe instances

- Properties associated with a class describe the attributes and relationships of the instances of the class



Individuals

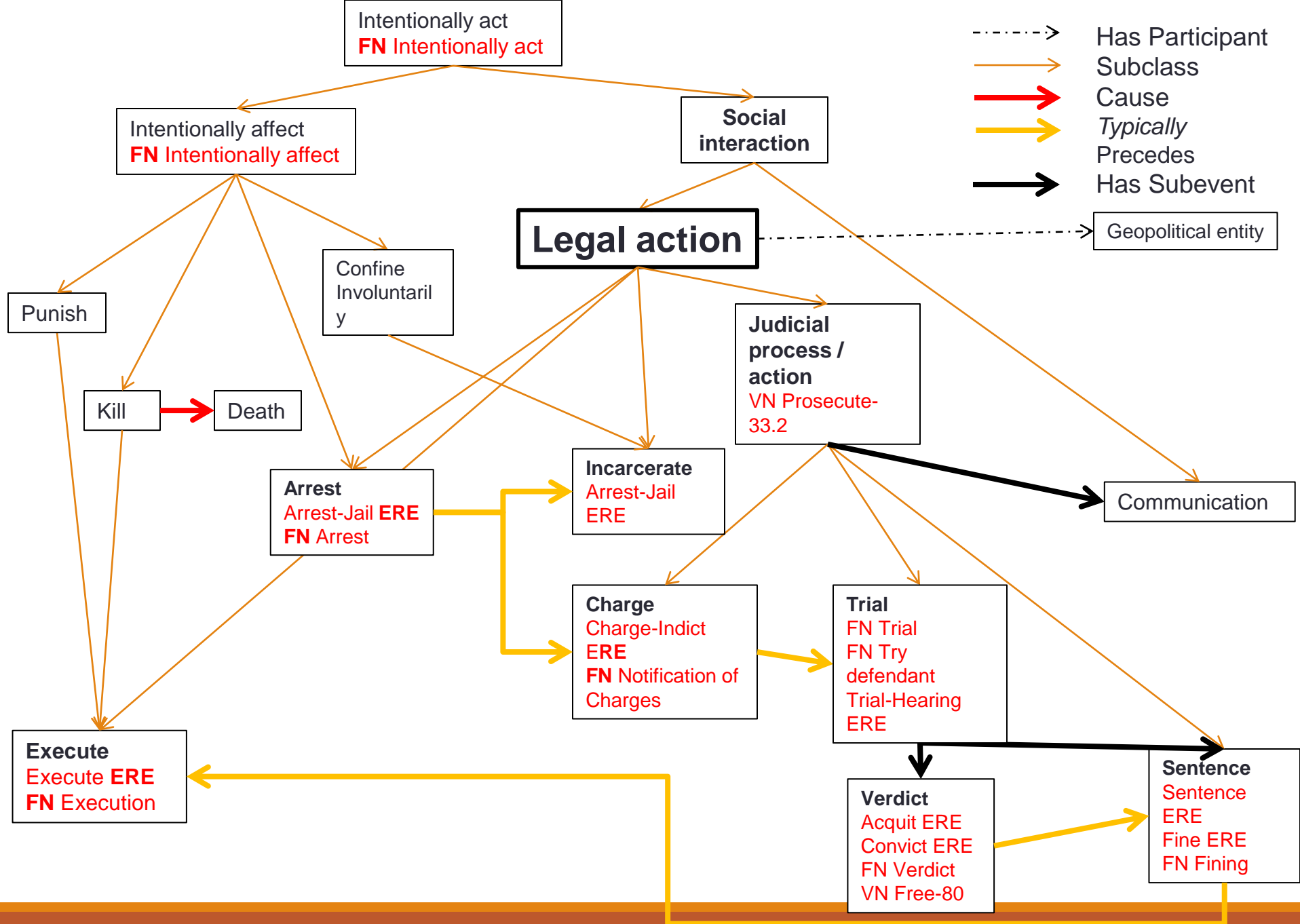
- Individuals are the last level of the ontology; they cannot be further specified
- They represent a materialization of the descriptions at the class level
- This is the level at which the actual data is put in
- The data depends on the application
 - grocery app?
 - tracking terrorist organizations?

Ontologies for NLP

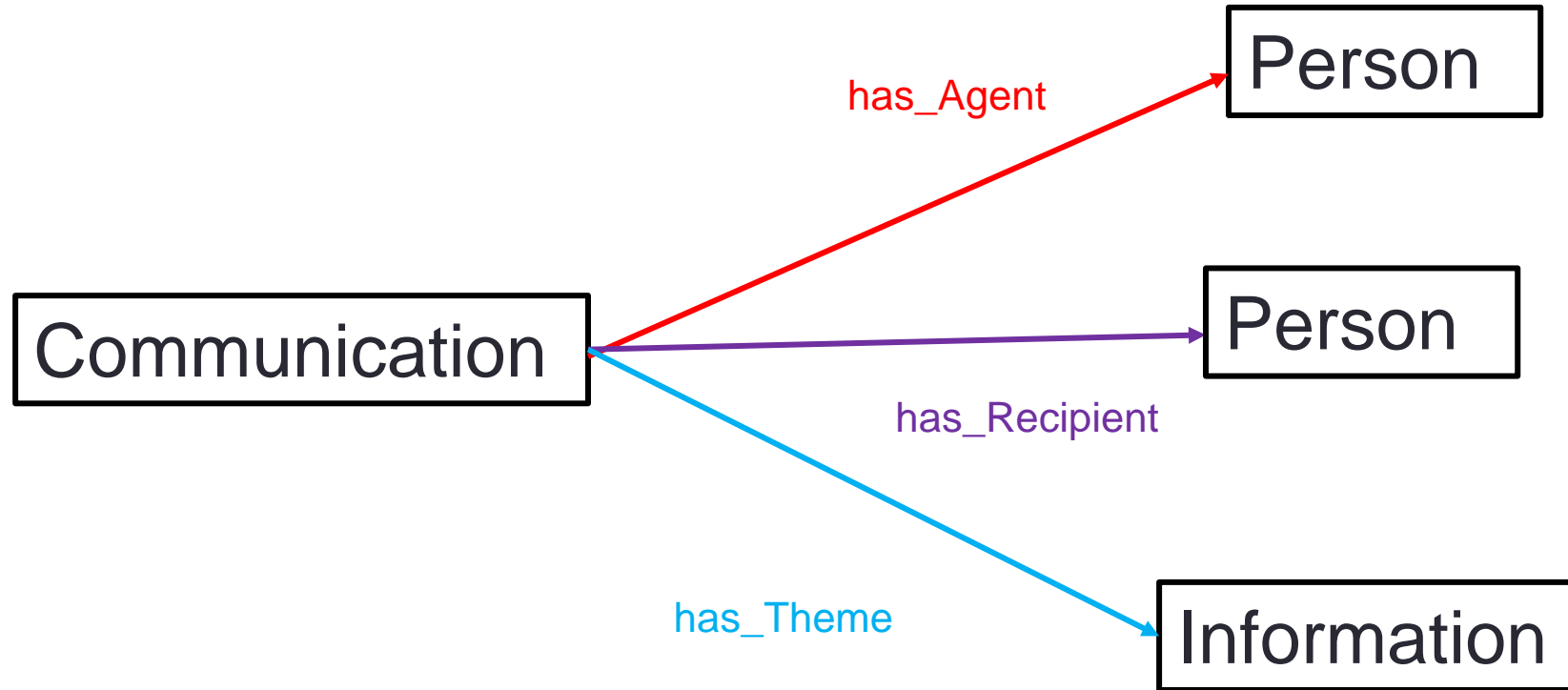
- Move from words to concepts/entities underlying the words
- A conceptual ontology with links to lexical items
- Bio-NLP
- Event extraction and participant tracking

Events in ontologies

- Events difficult to model in an ontology
 - is-a relations tricky to determine (killing, crime, murder, death)
 - where does an event start and end? (surgical event)
- Usually represented as **relations** between entities
 - relations can't have links to lexical items
 - relations can't have individuals (you might want to make lexical items the individuals or instances in annotation)
- Existing ontologies have **shallow** models of events
 - WordNet
 - SUMO (Suggested Upper Merged Ontology)



Event-Object Relations



Creation

Label	ArtifactExistence.Creation
Description	The act of creation or invention in which an entirely novel and unique physical or informational entity (or event) is formed for the first time from raw materials or components, either intentionally or through a causative event

Slot Role	Slot Argument Constraints
Creator	per, org, gpe, sid, event
Thing created	abs, fac, com, veh, wea, pth, inf, event?
Components/Materials	com, nat
Place	fac, loc, gpe

Temporal	
Start and End	(times specific to event)
Duration	1 minute through multiple years

Wear

Label	Wear (new social behavior top level?)
Description	Bearing or having clothing or other objects on the person

Slot Role	Slot Argument Constraints
Wearer	per
Thing worn	com
Body_Location	bod
Place	fac, loc, gpe

Temporal	
Start and End	(times specific to event)
Duration	1 minute through multiple years

Sanitize

Label	Sanitize
Description	Rendering pathogens harmless through methods including use of heat, antiseptics and antibacterial agents

Slot Role	Slot Argument Constraints
Agent/Sanitizer	per, org, gpe, sid
Sanitized object	fac, com, veh, wea
Sanitizing substance	com, nat
Pathogen	pth
Place	fac, loc, gpe

Temporal	
Start and End	(times specific to event)
Duration	1 minute through multiple years

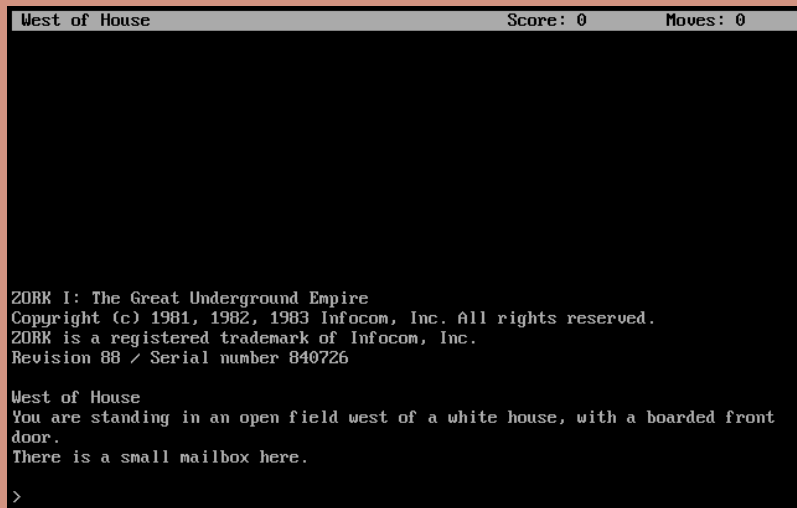
What is the problem with this approach?

- What might happen

The background is a solid blue gradient. Overlaid on this are numerous thin, white, curved lines that flow from the left side towards the right, creating a sense of movement and depth. These lines are more densely packed in some areas, forming a wave-like pattern that peaks towards the right side of the frame.

THE STORY SO FAR... (RECAP)

Interactive Fiction & Storytelling



Zork I



Façade, <https://www.playablstudios.com/facade>
https://cdn.download-free-games.com/cf/images/nfe/screens/facade_2_m.jpg



Sentient Beings
<https://grizel.itch.io/sentient-beings>

Old-School Interactive Fiction

Full Action
Space

Limited
Action
Space

Tabletop
Roleplaying
Games

Choose-Your-Own
Adventures

WHAT MAKES A STORY "GOOD"?



PollEv.com/laramartin527

What makes a story "good"?

Coherent

coherence

clear logic

coherent plot lines

consistency/continuity

Fun (diverse) but logical.

Interesting

surprises

interesting, have a surprising ending

compelling conflict

engaging narrative

convoluted

coherent, has an element of surprise, complex characters, beautiful worldbuilding

Relatable Characters

Compelling/relatable characters

character growth

Relatability

Compelling plot, interesting and relatable characters, humor, unexpected but properly explained plot points

compelling action and characters

decent storyline, compelling characters and good writing

Something innate in us?

I know it when I see it

Not everything written explicitly

A good story make me want to come back and leaves room for the reader to think and come to their own conclusions

Complexity/Theme

Underlying ideas/themes

Multiple plot elements

underlying deep / philosophical themes

Satisfying to read, gives interesting insights



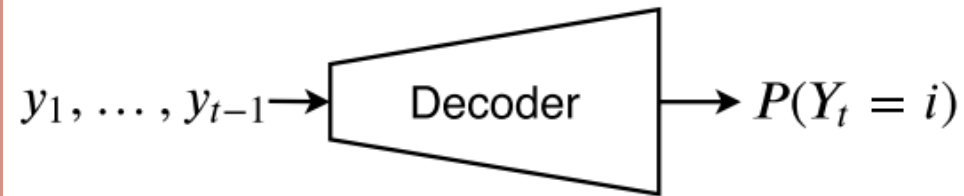
The background is a solid blue gradient. Overlaid on this are numerous thin, white, curved lines that flow from the left side towards the right, creating a sense of movement and depth. These lines are more densely packed in some areas, forming wave-like shapes that rise and fall across the frame.

NEURAL SYSTEMS

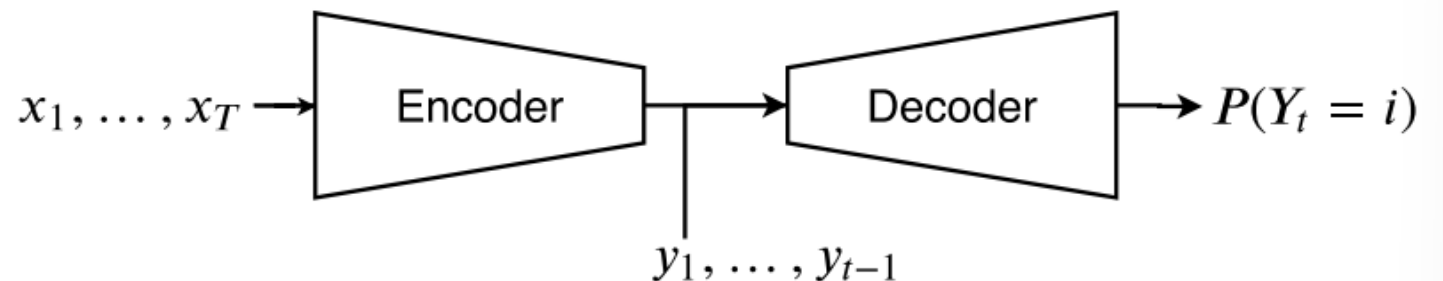
Neural Generation

- Probabilistic
 - Unconditioned $P(Y)$
 - Conditioned $P(Y|X)$

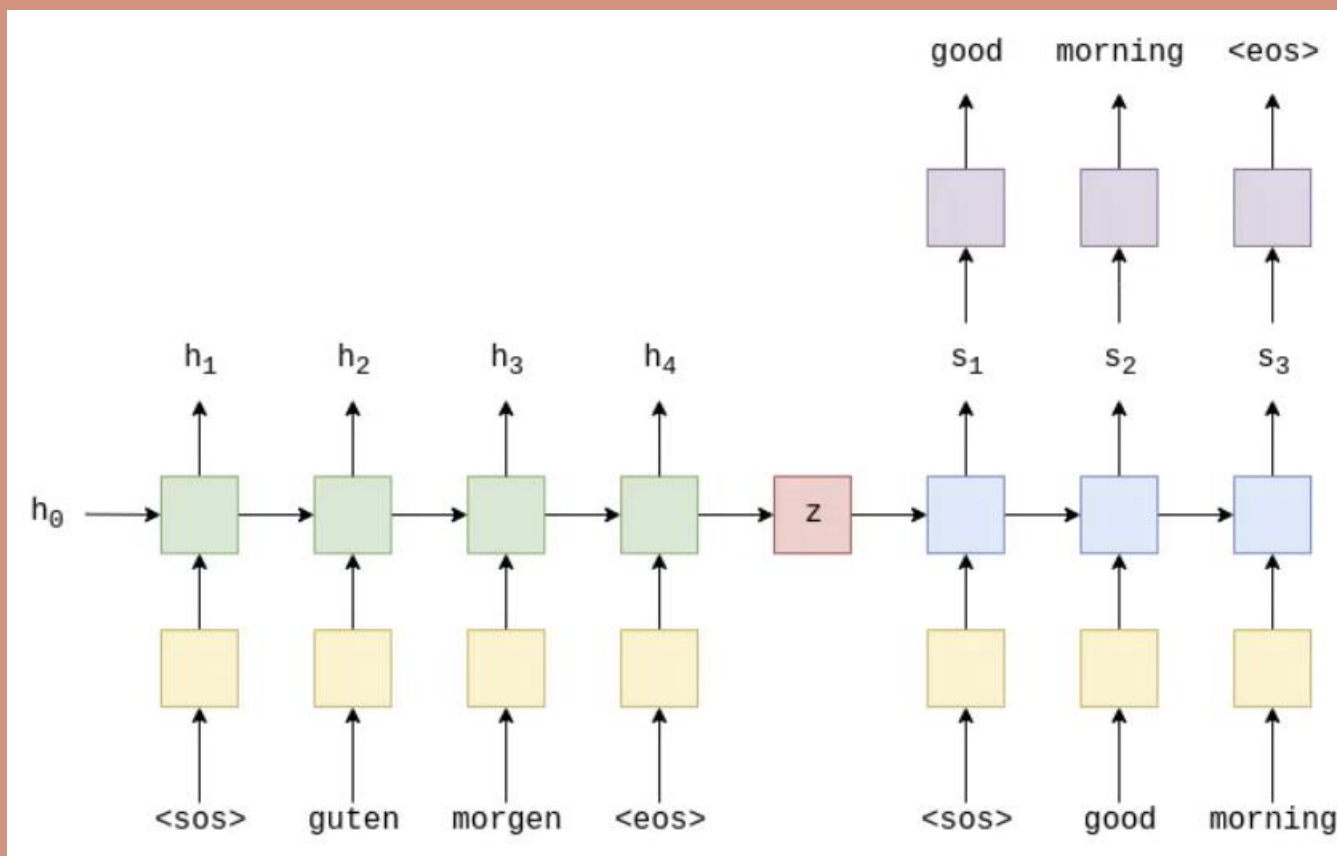
Unconditioned Language Model



Conditioned Language Model



RNNs (Sequence-to-Sequence)

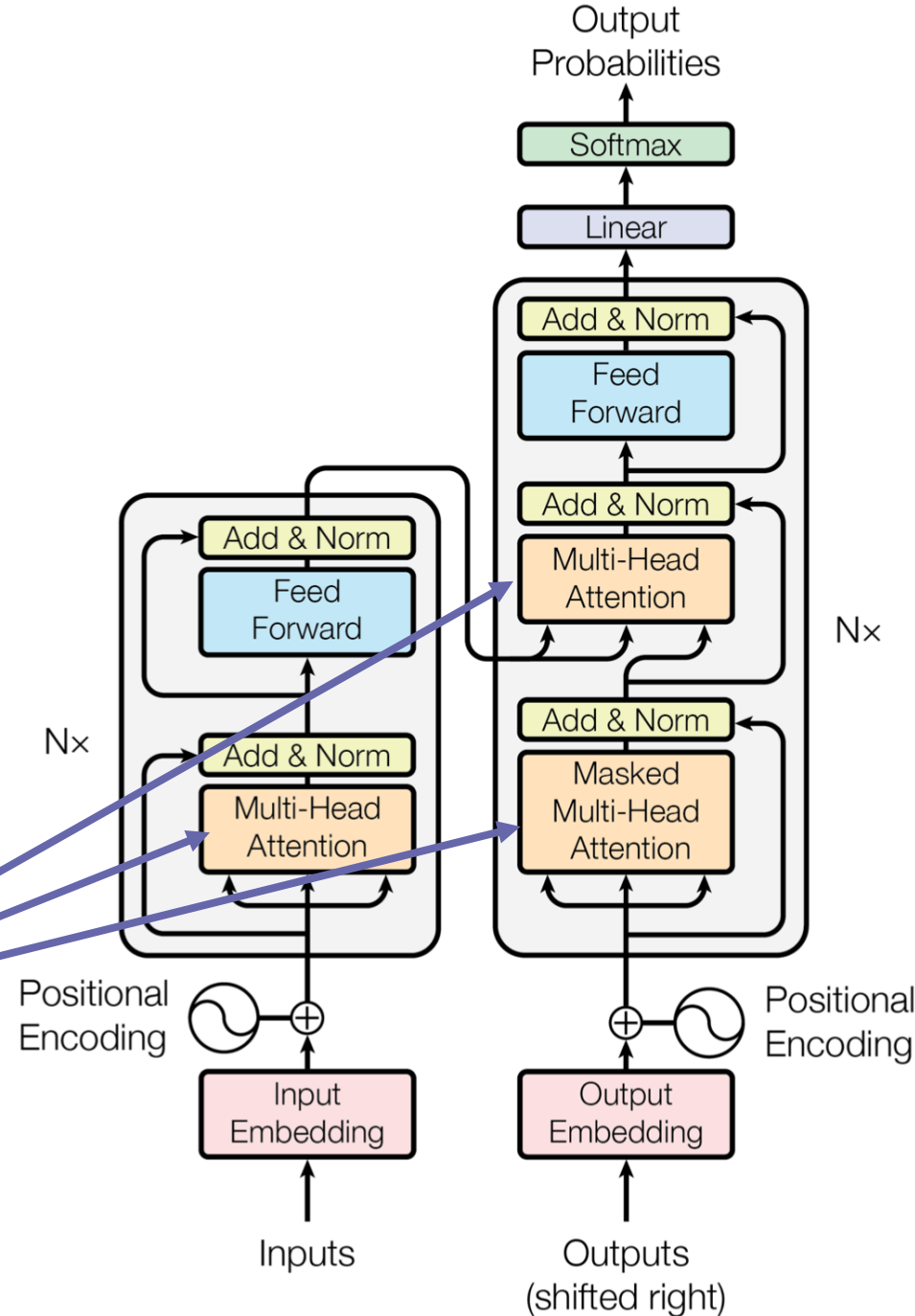


Transformers

Query Q – what you're "searching" for
Key K – what you compare the query against
Value V – the results that is paired to the key

Attention is All You Need!

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)v$$



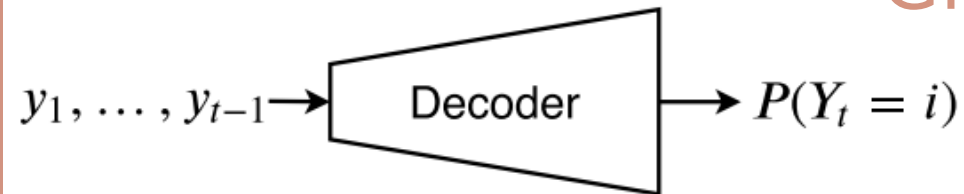
Transformer Types

Encoder-Only:
BERTs

What are encoder-only
models useful for?

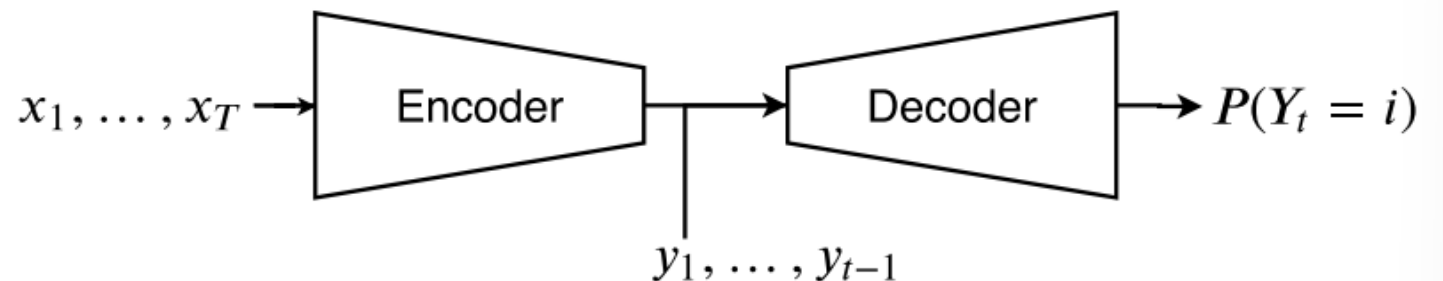
Unconditioned Language Model

GPTs



Conditioned Language Model

T5s



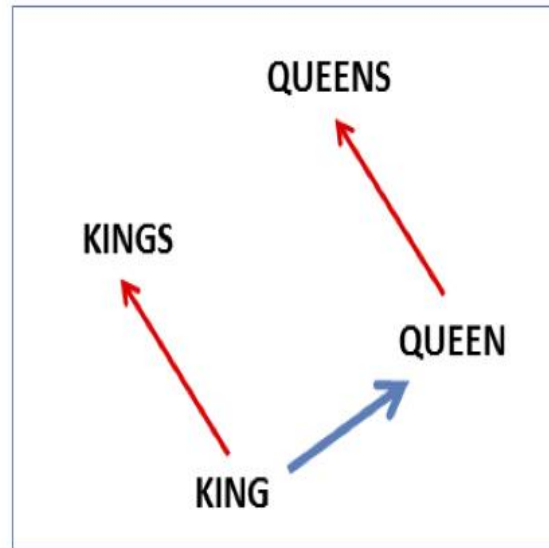
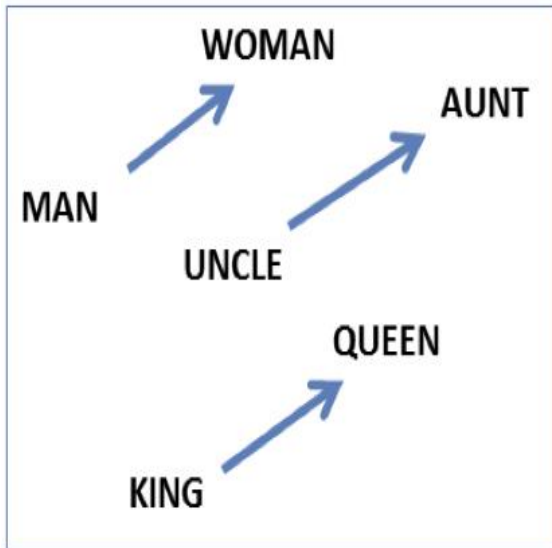
(Some) Properties of Embeddings

Why are embeddings useful for neural networks?

Capture “like” (similar) words

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

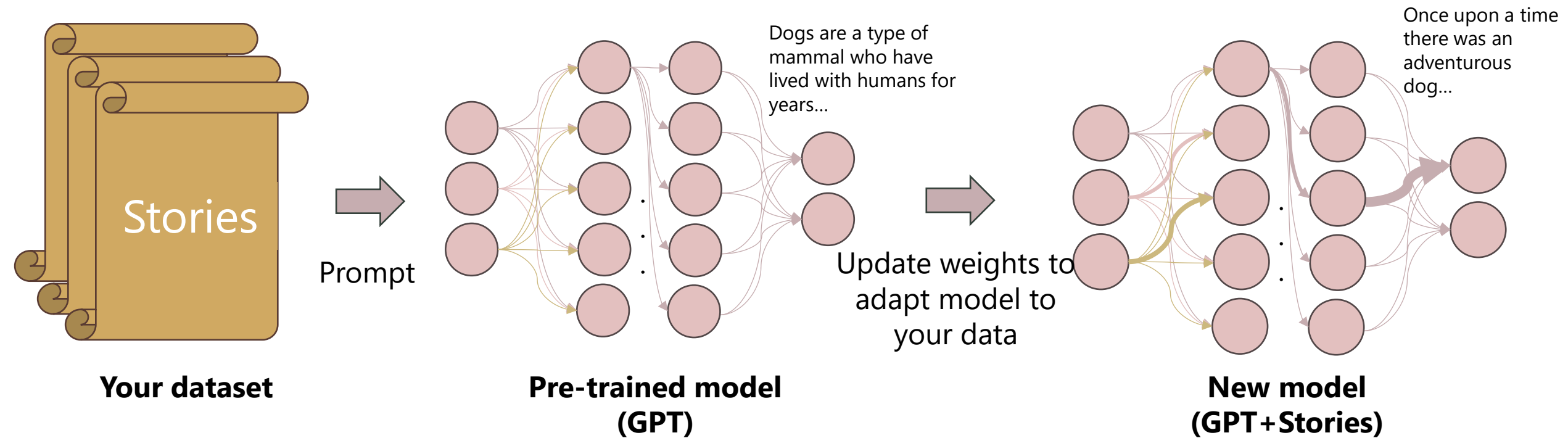
Capture relationships



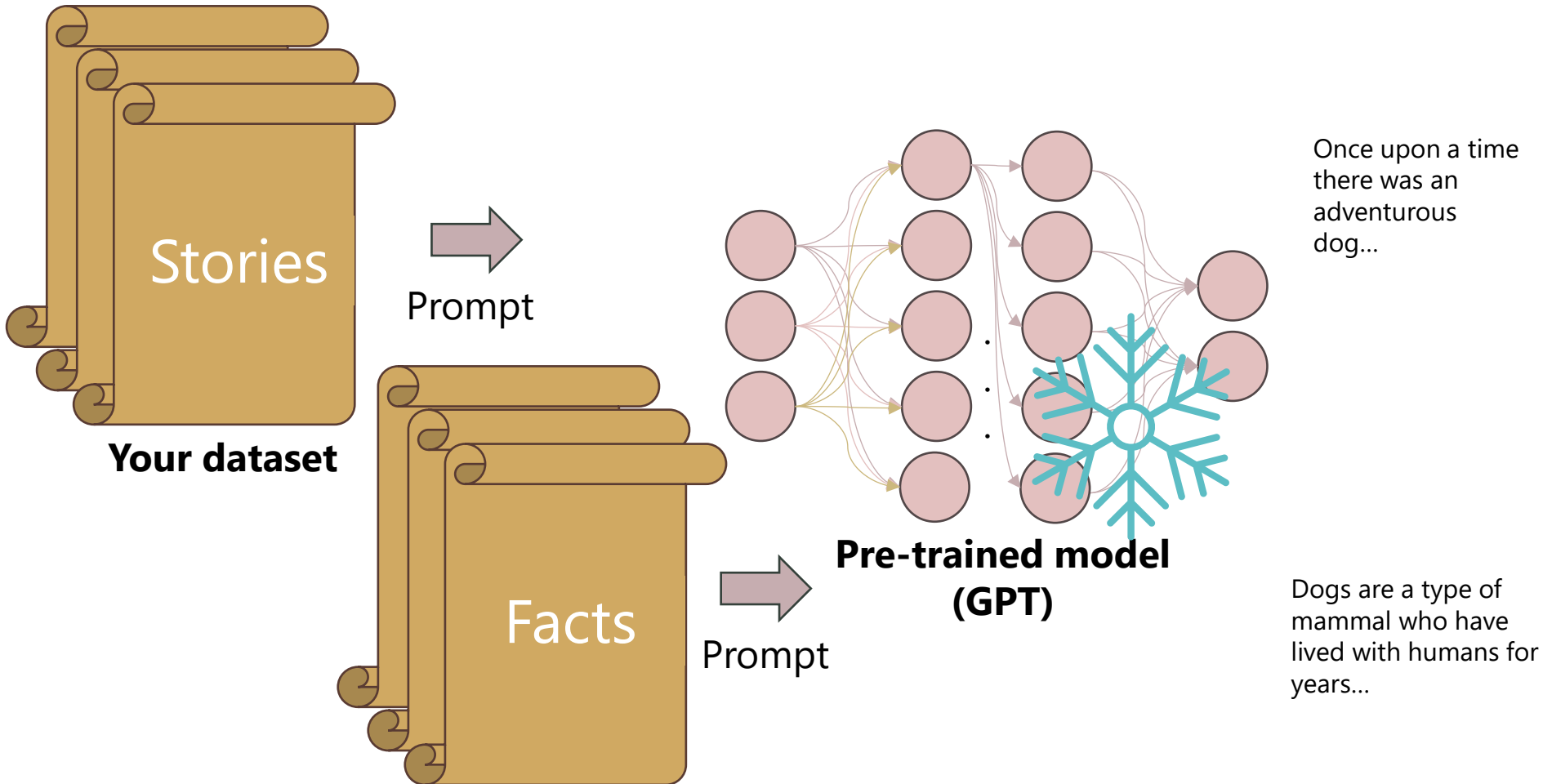
$$\text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')}$$

$$\text{vector('Paris')} - \text{vector('France')} + \text{vector('Italy')} \approx \text{vector('Rome')}$$

Finetuning



Prompting

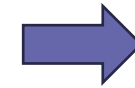
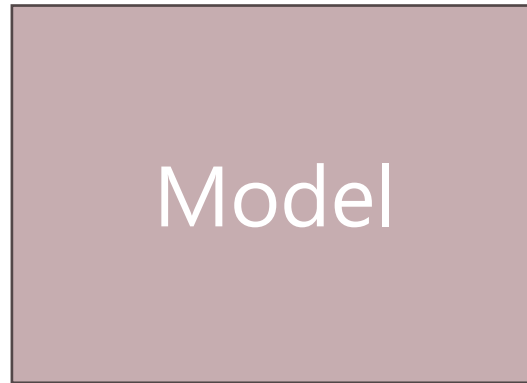
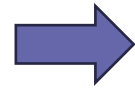


Zero-shot Prompting

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

Instructions

Task



Output

Sentence:
The dog ate the giant fish.

Few-shot Prompting

Instructions

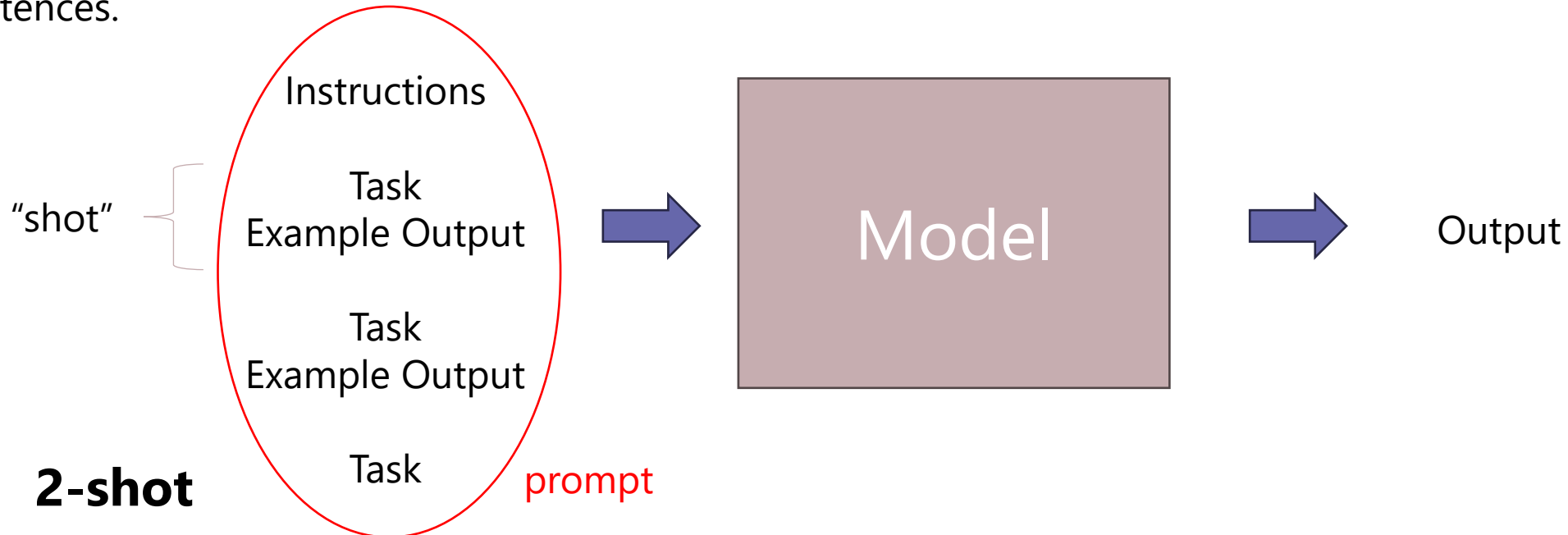
You are a helpful assistant.
You will be tagging the
parts of speech in
sentences.

Task

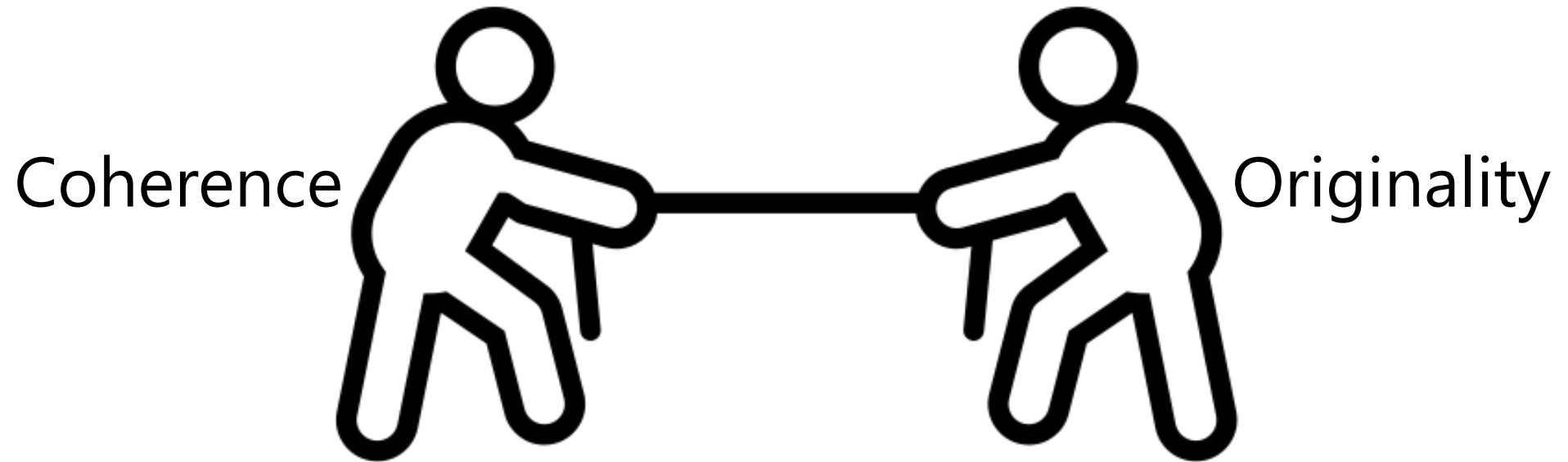
Sentence:
The dog ate the giant fish.

Example Output

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



Lara's Language Model Tradeoff



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

Neural Story Generation

The hungry dog licked her lips as she watched her owner eat.

"You've been a good girl," he told her. "I think you deserve a reward."

On Theme



Once she was done, she jumped back on the couch and waited patiently.

Her owner took a piece of steak out of the fridge and gave it to her.

Grammar



"Thank you," he said. "I'm glad you're my dog."

Remembering
Story State



She wagged her tail and ate the steak.

"If you're good, you can have a treat later," he said. "But for now, you have to sleep. I have a long day tomorrow."

Commonsense
Reasoning



She nodded and lay down on the floor.

Her owner got up, turned off the lights, and lay down on the bed.

HW 2

Homework 2: Prompting and Fine-tuning

In this homework, we're going to use OpenAI's API to generate text adventure game components automatically. Starting with the [prompting ideas from class](#) and [generating descriptions using the Playground](#), we'll show how to finetune models to perform specific tasks. In particular, you will generate room descriptions and item properties for text adventure games.

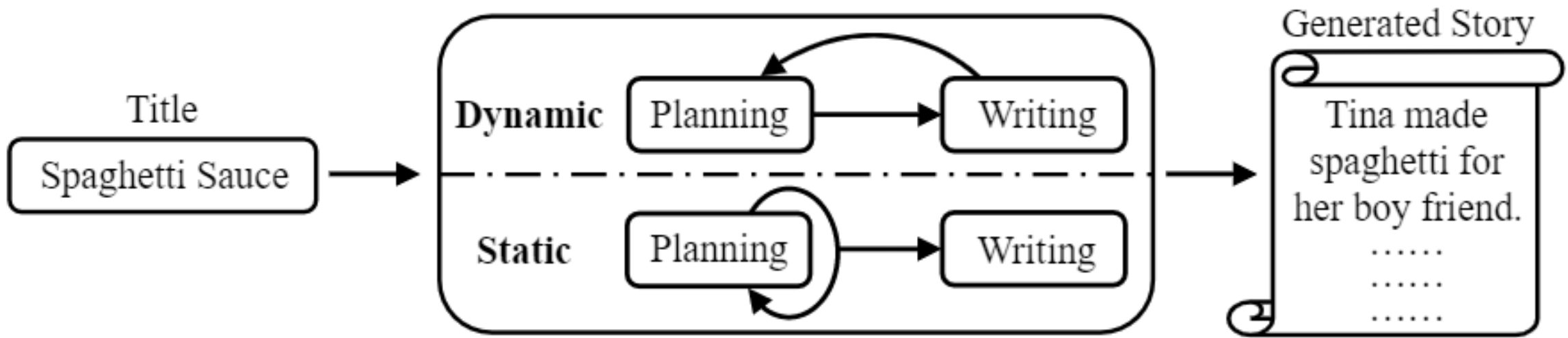
Learning Objectives

For this assignment, we will check your ability to:

- Use the OpenAI API for few-shot prompting GPT models
- Use the OpenAI API for finetuning GPT early models
- Setup data for finetuning
- Compare early finetuned output to modern few-shot output

Guided Neural Story Generation

Integrating ways of including structure



Re³

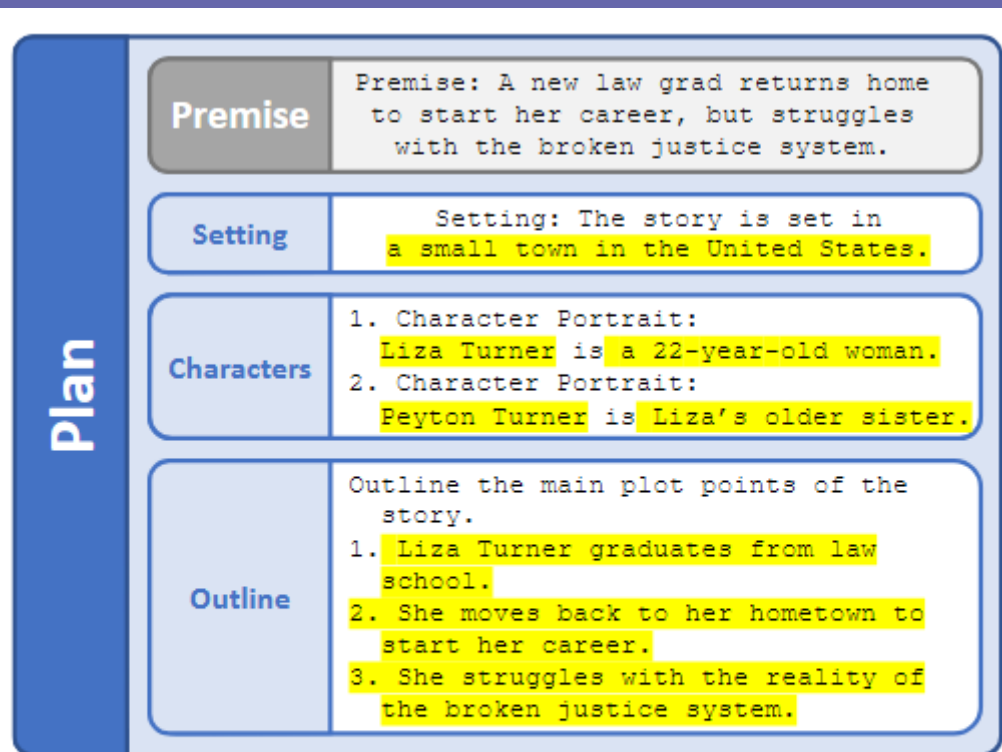


Figure 2: Illustration of Re³'s Plan module, which prompts a language model to generate a setting, characters, and outline based on the premise. Highlighting indicates generated text.

Re³: Generating Longer Stories With Recursive Reprompting and Revision

Kevin Yang¹ Yuandong Tian² Nanyun Peng³ Dan Klein¹

¹UC Berkeley, ²Meta AI, ³UCLA

{yangk,klein}@berkeley.edu,yuandong@meta.com,violetpeng@cs.ucla.edu

Abstract

We consider the problem of automatically generating longer stories of over two thousand words. Compared to prior work on shorter stories, long-range plot coherence and relevance are more central challenges here. We propose the Recursive Reprompting and Revision framework (Re³) to address these challenges by (a) prompting a general-purpose language model to construct a structured overarching plan, and (b) generating story passages by repeatedly injecting contextual information from both the plan and current story state into a language model prompt. We then revise by (c) reranking different continuations for plot coherence and premise relevance, and finally (d) editing the best continuation for factual consistency. Compared to similar-length stories generated directly from the same base model, human evaluators judged substantially more of Re³'s stories as having a coherent overarching plot (by 14% absolute increase), and relevant to the given initial premise (by 20%).

1 Introduction

Generating long-term coherent stories is a long-standing challenge for artificial intelligence, requiring

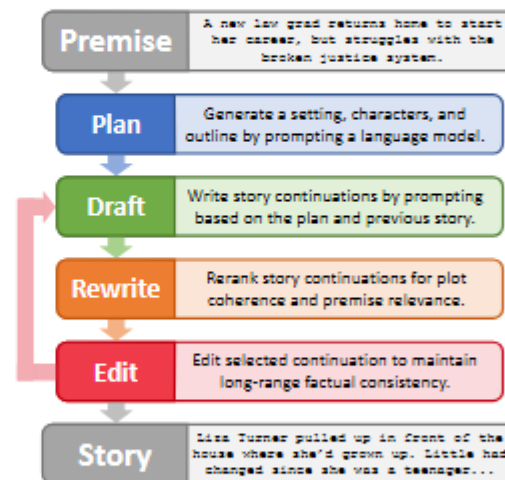


Figure 1: High-level overview of Re³.

length increases limited primarily by evaluation rather than technical issues.¹ Generating stories of such length faces qualitatively new challenges compared to prior work on shorter stories. First, the system must maintain a coherent overarching plot over thousands of words. Given an initial premise, it should maintain relevance to this premise over thousands of words as well. Additional challenges include preservation of narration style and avoiding

HW 3

Homework 3: Guided Generation

Now that you know how to prompt an LLM from HW2, we will be using some guided story generation techniques from Module 2. In this homework, you will be following a generation pipeline inspired by the [Plan-and-Write system](#). In their work, they generated keywords from a title and then generated a story from the keywords. They tried both dynamic and static schemas to integrate the planning into their generation pipeline. This homework will focus on the “static” schema but use a pre-trained LLM instead of their RNN model.

Learning Objectives

For this assignment, we will check your ability to:

- Prompt an LLM to generate stories given varying amounts of context
- Implement NLP evaluation metrics using existing libraries
- Compare the quality of guided vs unguided story generation
- Determine the adequacy of automated metrics like BLEU and ROGUE for creative evaluation

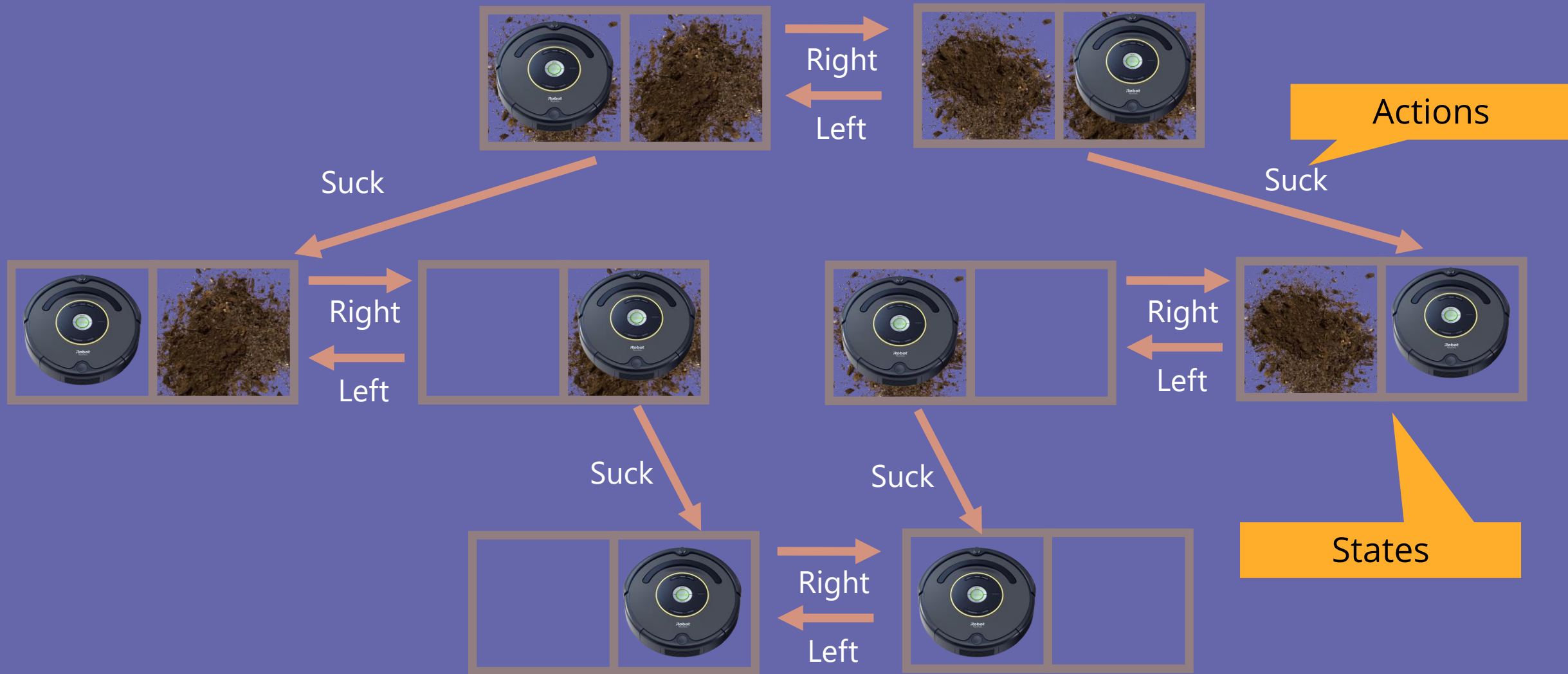
The background of the slide is a solid blue gradient. Overlaid on this are numerous thin, white, curved lines that flow from the left side towards the right, creating a sense of movement and depth. These lines are more densely packed in some areas, forming a wave-like pattern that peaks towards the right side of the slide.

SYMBOLIC SYSTEMS

Symbolic Systems: Planning

- Planning = *search* for a *plan*
- In story generation, this means we're looking for a plan where the goal is reached
- What's the goal? Depends on the story you're telling
 - E.g. Ending a conflict between characters, Robber steals from player character

Search



Search Strategies

Several classic search algorithms differ only by the order of how they expand their search trees

You can implement them by using different queue data structures

Depth-first search = LIFO queue

Breadth-first search = FIFO queue

Greedy best-first search or **A* search** = Priority queue

Action Castle Map Navigation

Let's consider the sub-task of navigating from one location to another.

Formulate the *search problem*

States: locations in the game

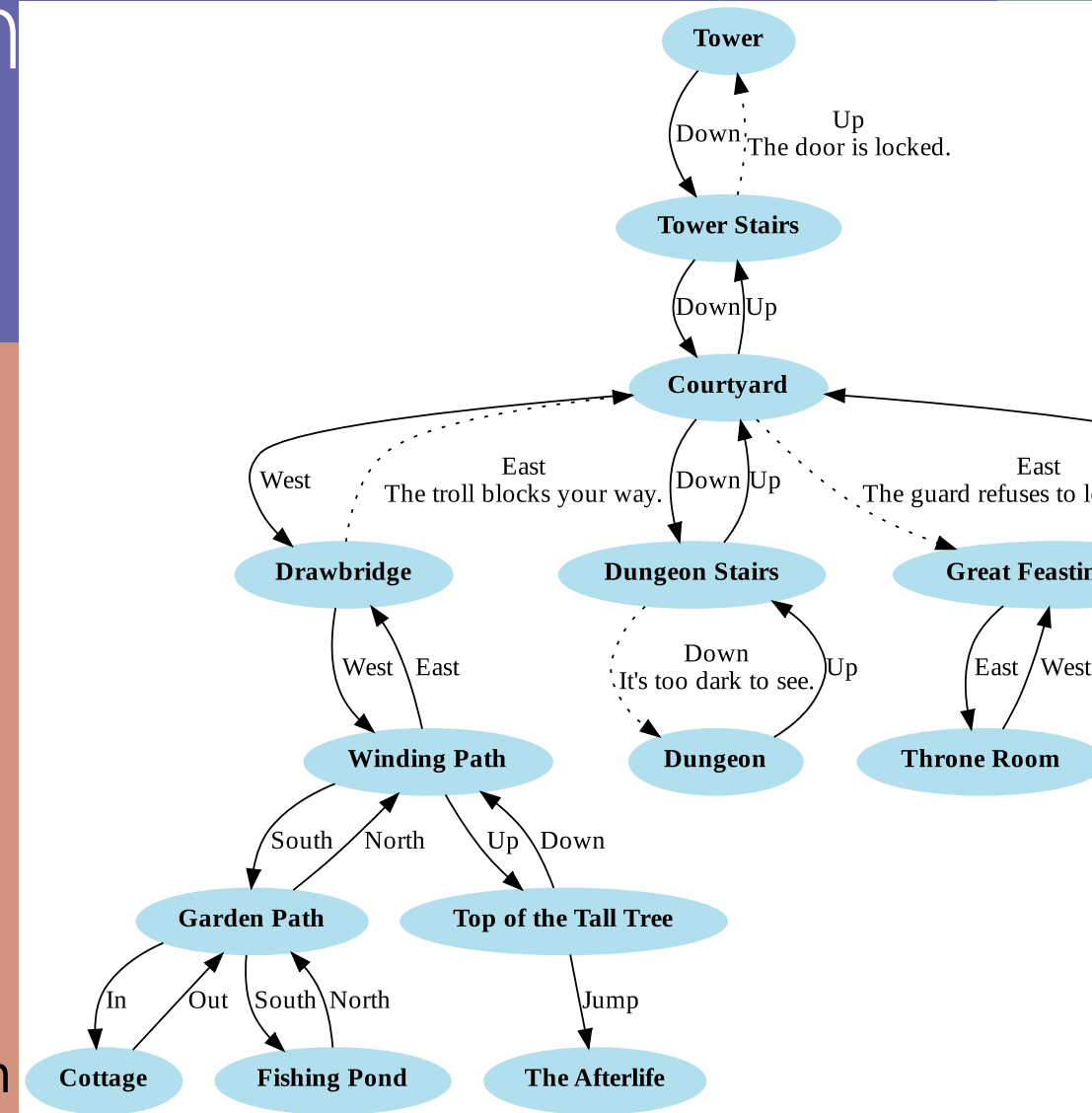
Actions: move between connected locations

Goal: move to a particular location like the **Throne Room**

Performance measure: minimize number of moves to arrive at the goal

Find a *solution*

Algorithm that returns sequence of actions to get from the start state to the goal.



What are we planning over?

Structure (Schemas)

KB Schemas

Scripts

Procedures

} Organization of Commonsense Knowledge

States

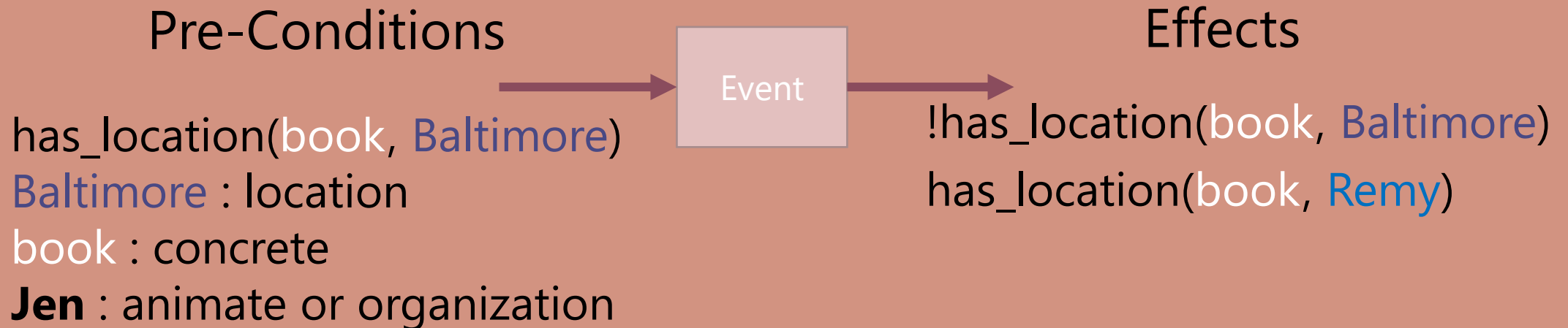
VerbNet Schema

Jen sent the book to Remy from Baltimore.

Baltimore : location
book : concrete
Jen : animate or organization
!has_location(book, Baltimore)
has_location(book, Remy)

Pre-Conditions and Effects

Jen sent the book to Remy from Baltimore.



What are we planning over?

Structure (Schemas)

KB Schemas

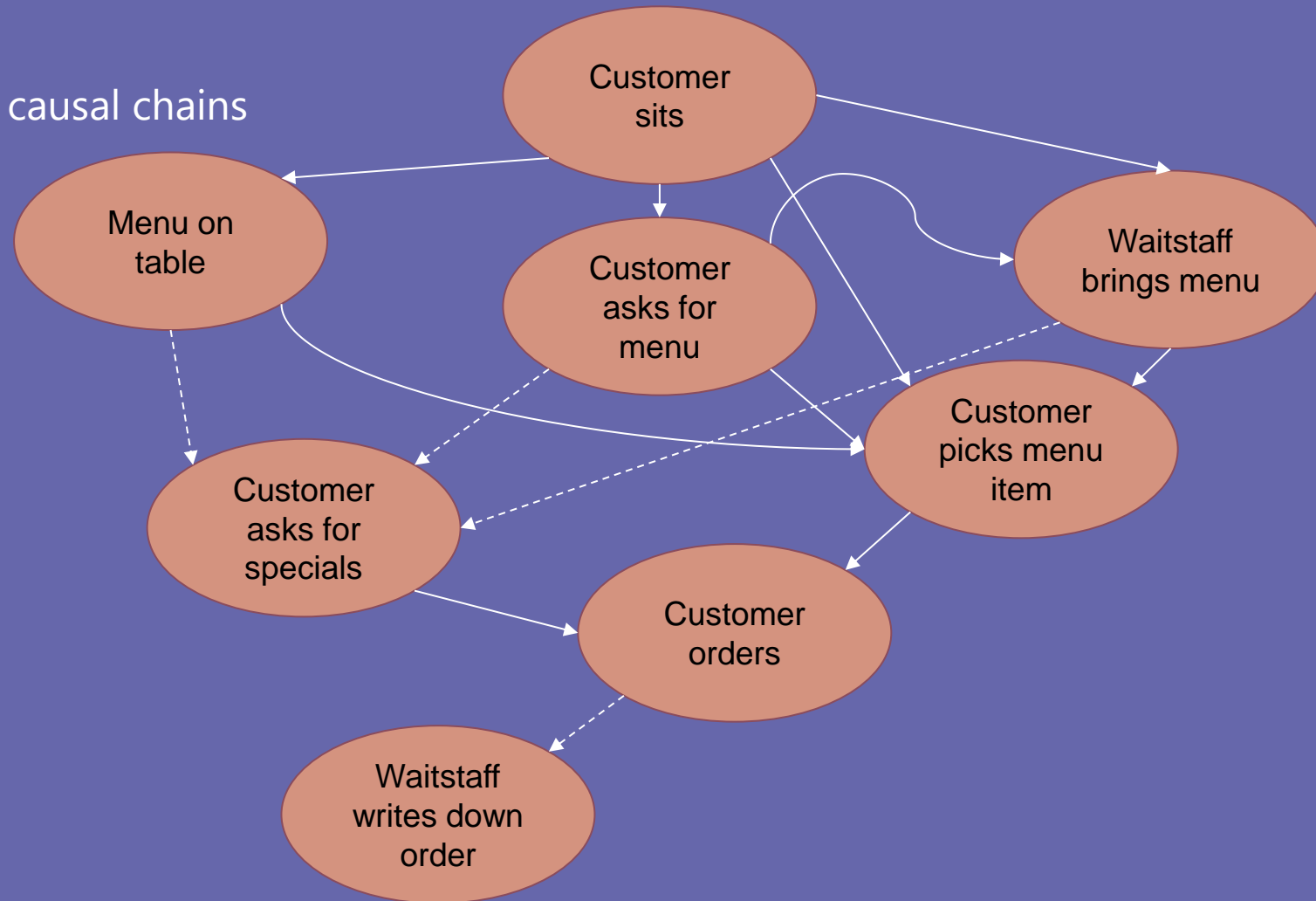
Scripts

Procedures

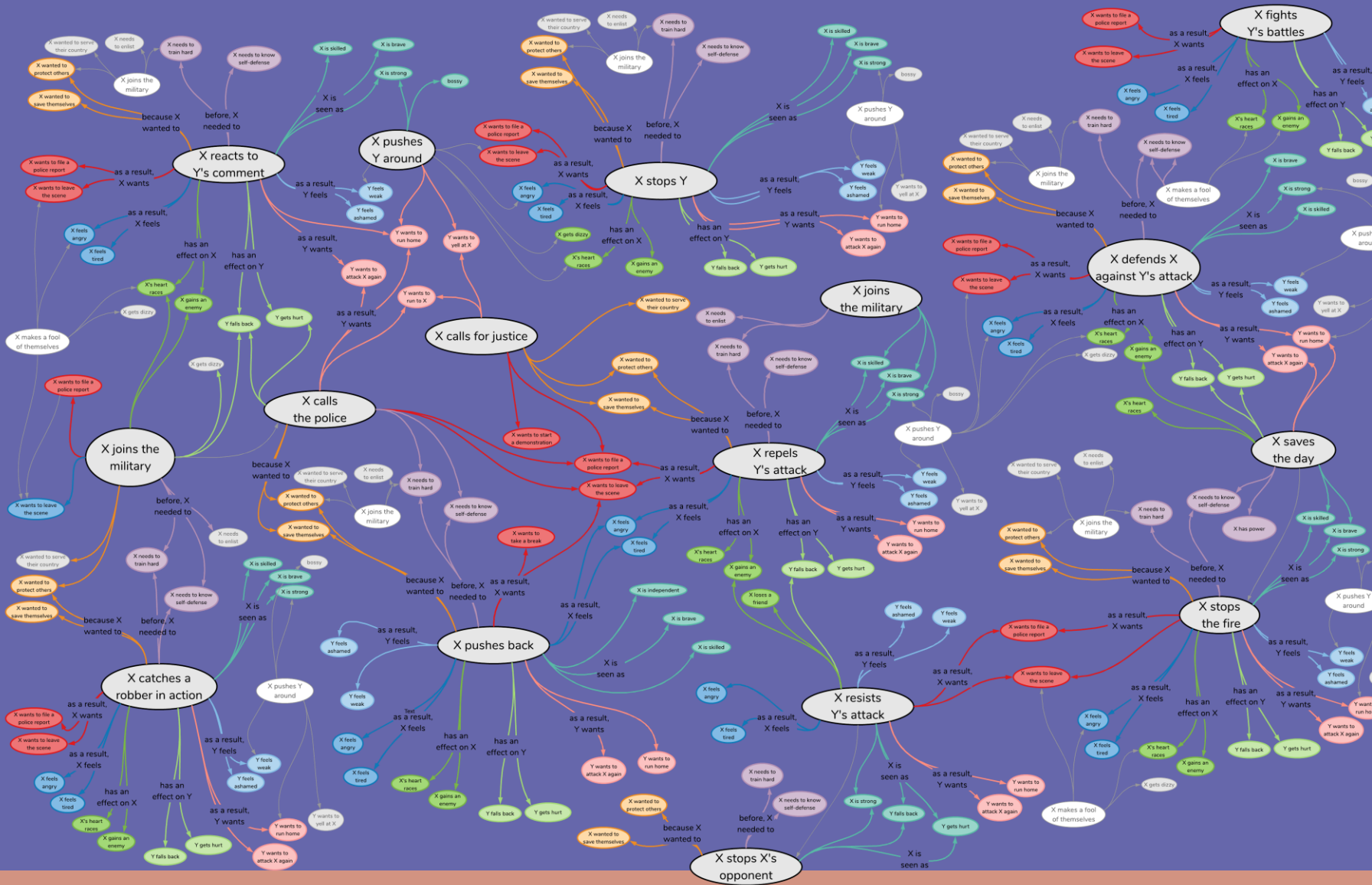
} Organization of Commonsense Knowledge

Scripts

Common sequence of causal chains



What's the difference between a script and a knowledge graph?



[PollEv.com/laramartin527](https://www.poll-ev.com/laramartin527)

Example of a Probabilistic Event Representation

From sentence, extract event representation:

(subject, verb, direct object, modifier, preposition)

Original sentence: yoda uses the force to take apart the platform

Events:

yoda use force Ø Ø

yoda take_apart platform Ø Ø

Generalized Events:

<PERSON>0 fit-54.3 power.n.01 Ø Ø

<PERSON>0 destroy-44 surface.n.01 Ø Ø

Procedures: Script with a goal

category

FOOD AND ENTERTAINING » DINING OUT

goal

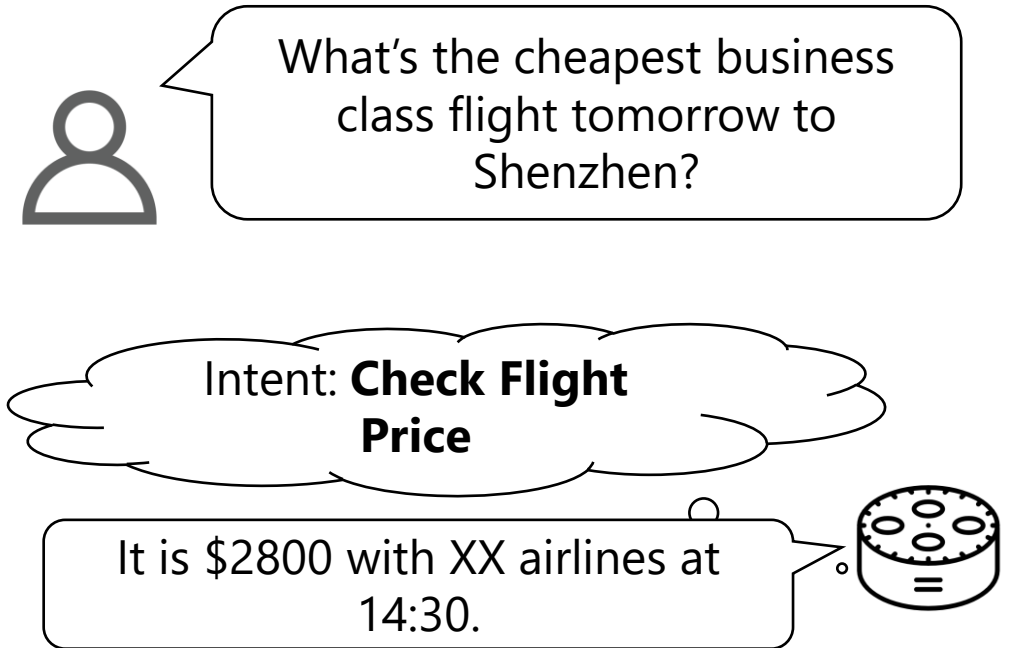
How to Eat at a Sit Down Restaurant

steps

- 1 Order drinks first.** If your server immediately asks you for your drinks and you're not sure, consider asking for water while you look over the drink menu. It's important not
- 2 Ask about daily specials.** Many restaurants will have rotating specials that can offer tasty surprises. Ask about the vegetable, fish, or soup of the day as well to make sure
- 3 Look over the menu and place your food order.** Usually, by the time that the server brings your beverages, you can begin to order an appetizer. This is where looking at

Intent Detection

- Task-oriented dialog systems needs to match an **utterance** to an **intent**, before making informed responses
- Sentence classification task
 - Given an utterance, and some candidate intents
 - Choose the correct intent



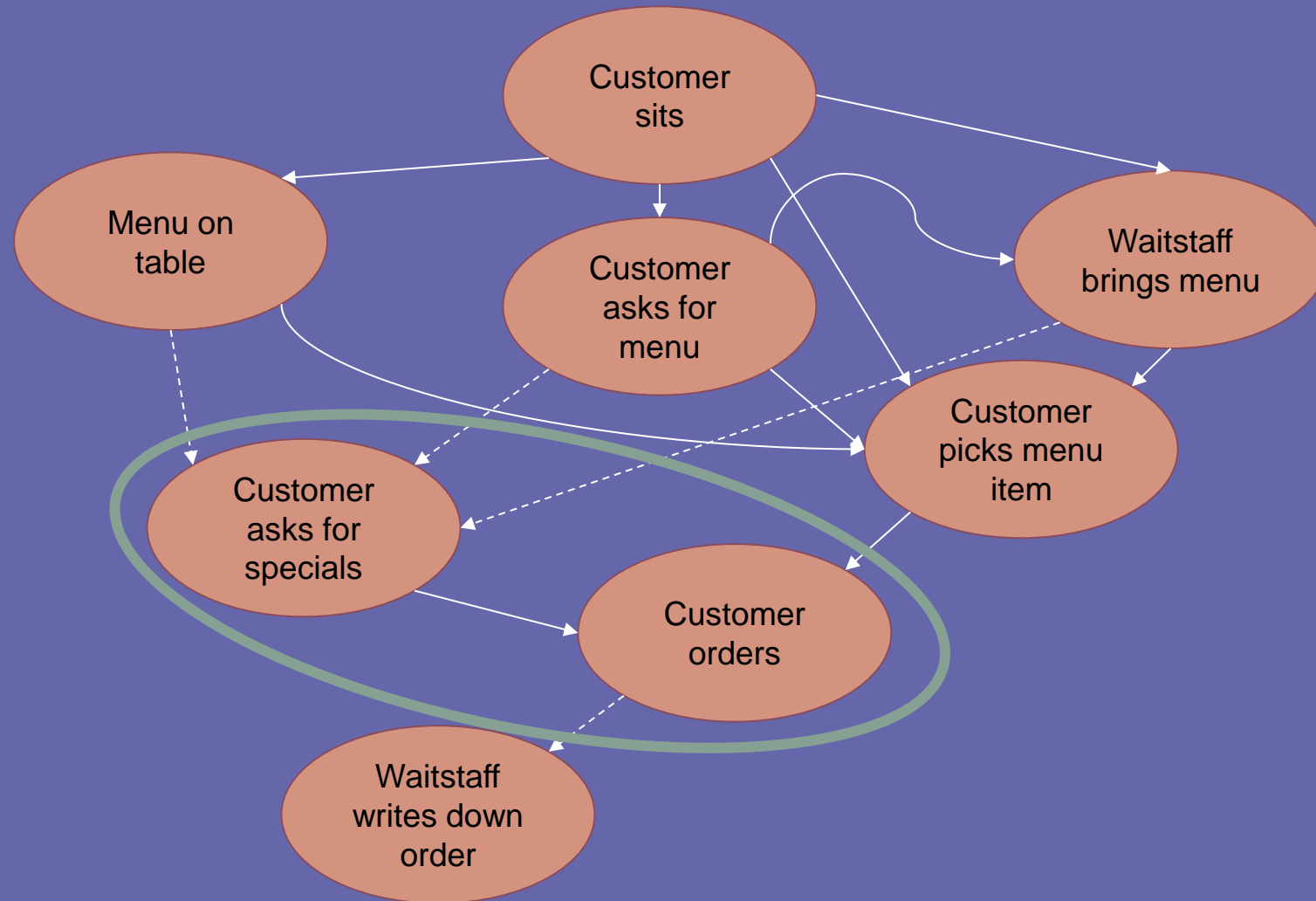
Example from Snips (Coucke et al., 2018)

Utterance: "Find the schedule at Star Theatres."

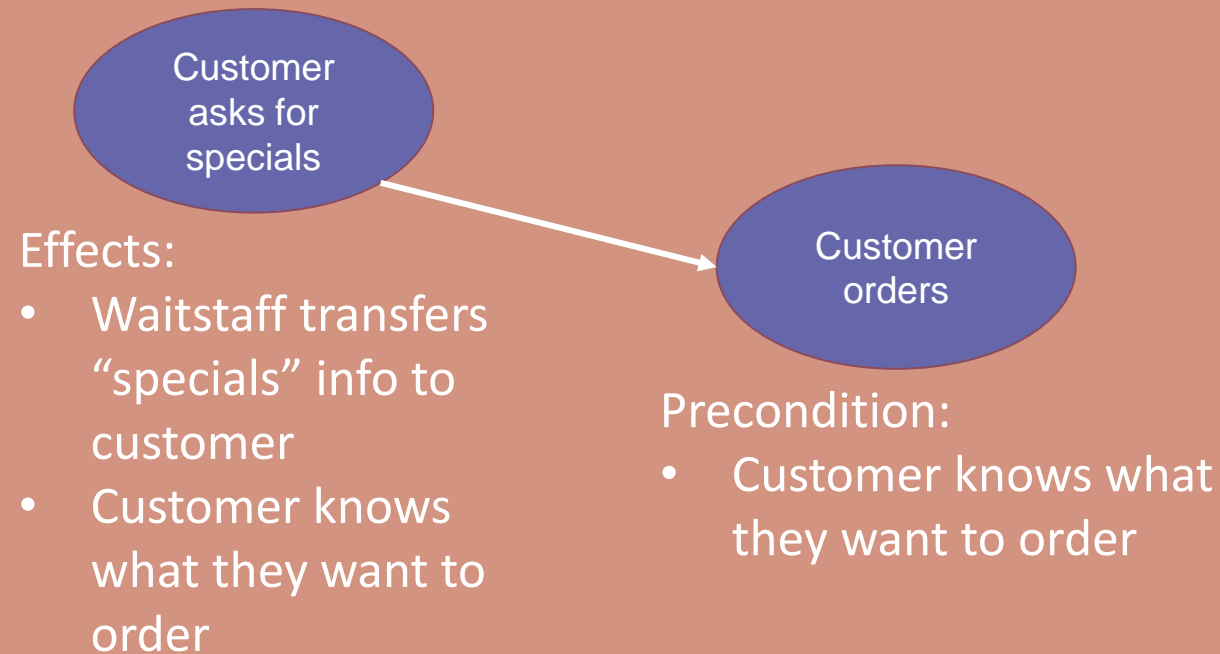
Candidate intents: Add to Playlist, Rate Book, Book Restaurant, Get Weather, Play Music, Search Creative Work, **Search Screening Event**

(Zhang et al., 2020): Intent Detection with WikiHow

Scripts



Causal Links



Causal Links \rightarrow Actions for Planning

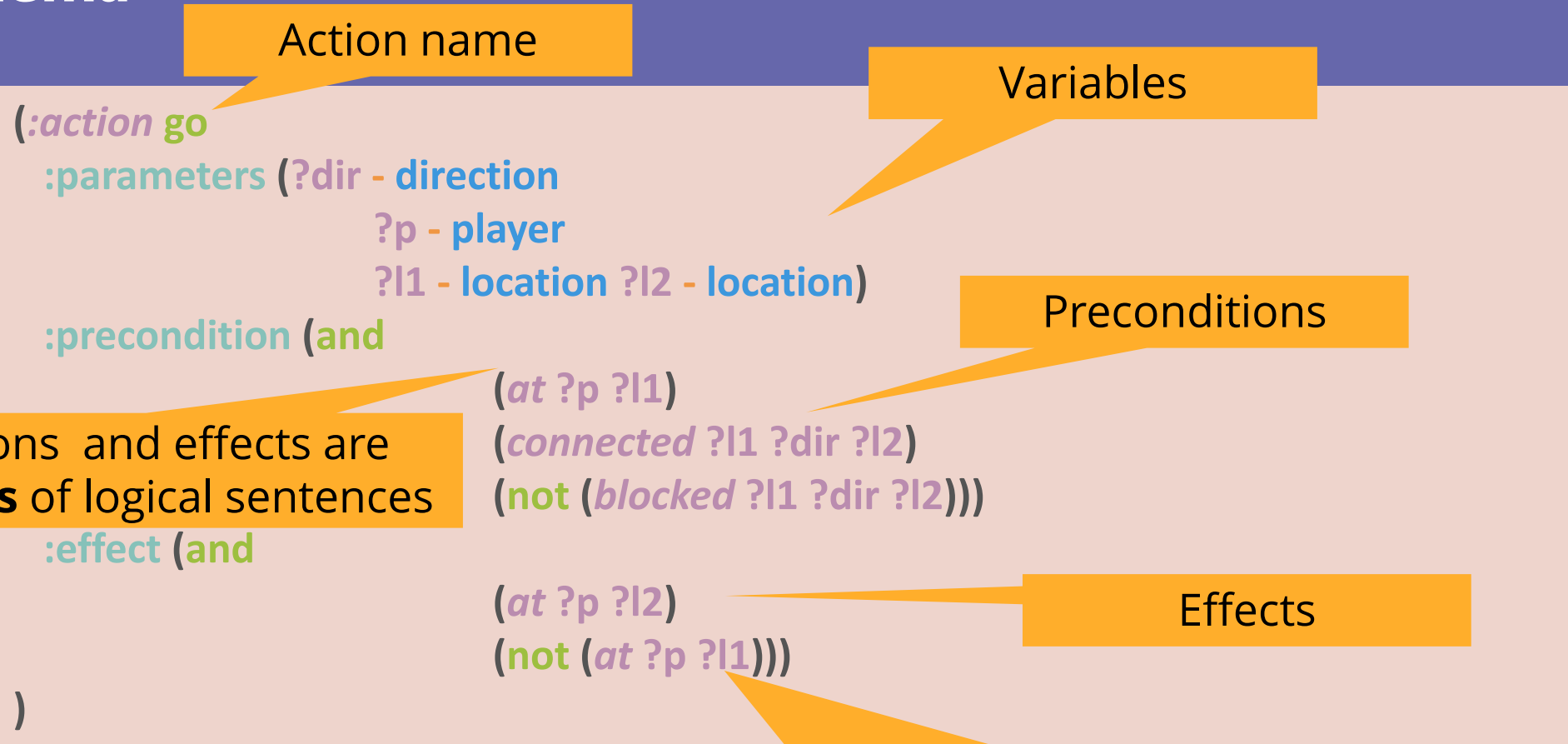
a: buy(Tom, Potion, Merchant, Market)

***PRE(a): at(Tom) = Market \wedge at(Merchant) = Market \wedge
at(Potion) = Merchant \wedge wealth(Tom) \geq 1***

***EFF(a): at(Potion) = Tom \wedge wealth(Merchant) $+=$ 1 \wedge
wealth(Tom) $-=$ 1***

Representation Language

Planning Domain Definition Language (PDDL) express **actions** as a **schema**



Preconditions and effects are **conjunctions** of logical sentences

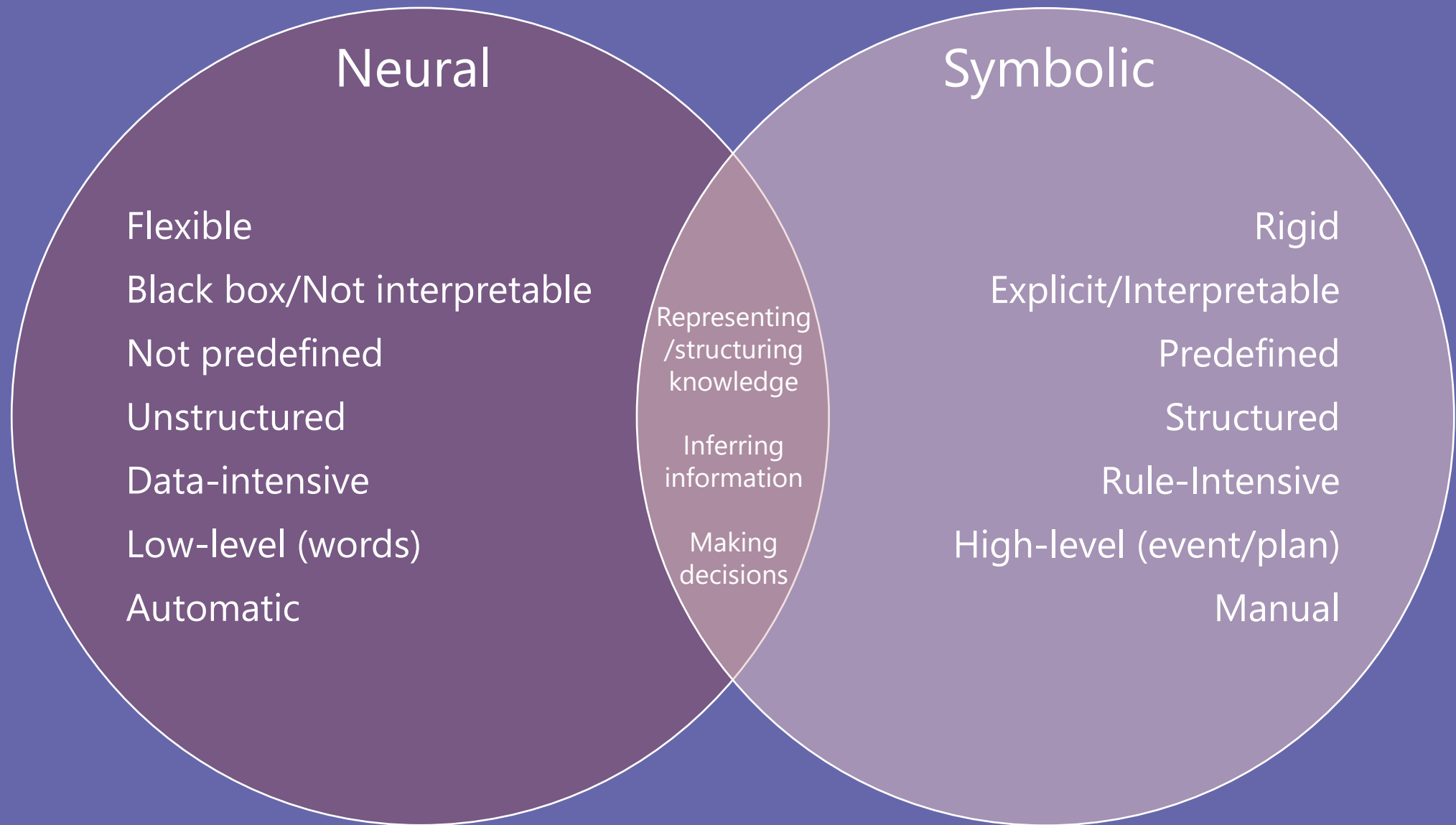
These logical sentences are **literals** – positive or negated atomic sentences

HW 4

Homework 4: Creating Sabre Problems

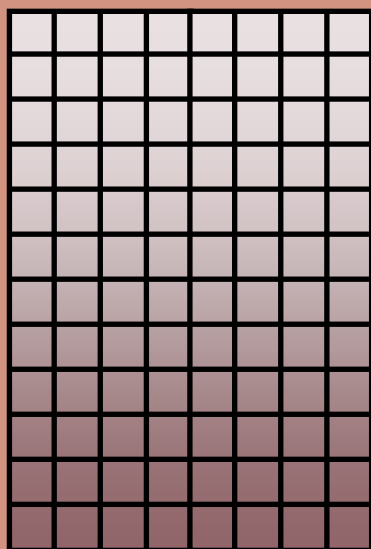
Learning Objectives

- Figure out how to write a problem for a planning program.
- Generate a planning problem using Github Copilot.
- Compare the processes of generating a planning problem by hand vs LLM

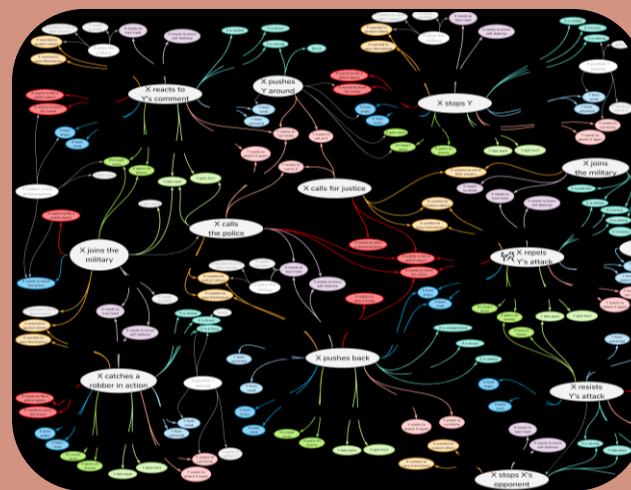
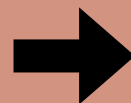


Solution Outline

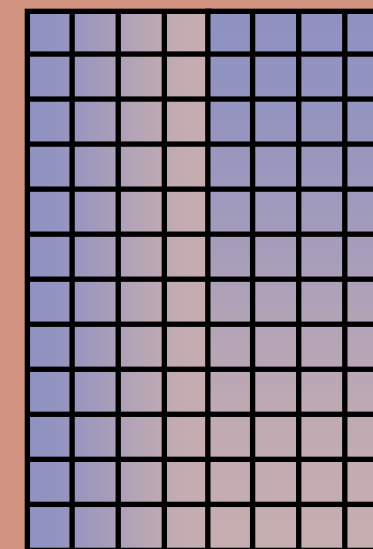
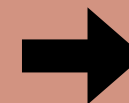
- Leverage manually curated commonsense knowledge resources
- Learn from the examples to induce new relationships
- Scale up using language resources



Learn word embeddings
from language corpus



Retrofit word embeddings
on semantic resource



Learn knowledge-
aware embeddings

Transfer Learning from Language

