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:

$$\exists e Liking(e) \land Liker(e, Franco) \land Liked(e, Frasca)$$

- 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.*

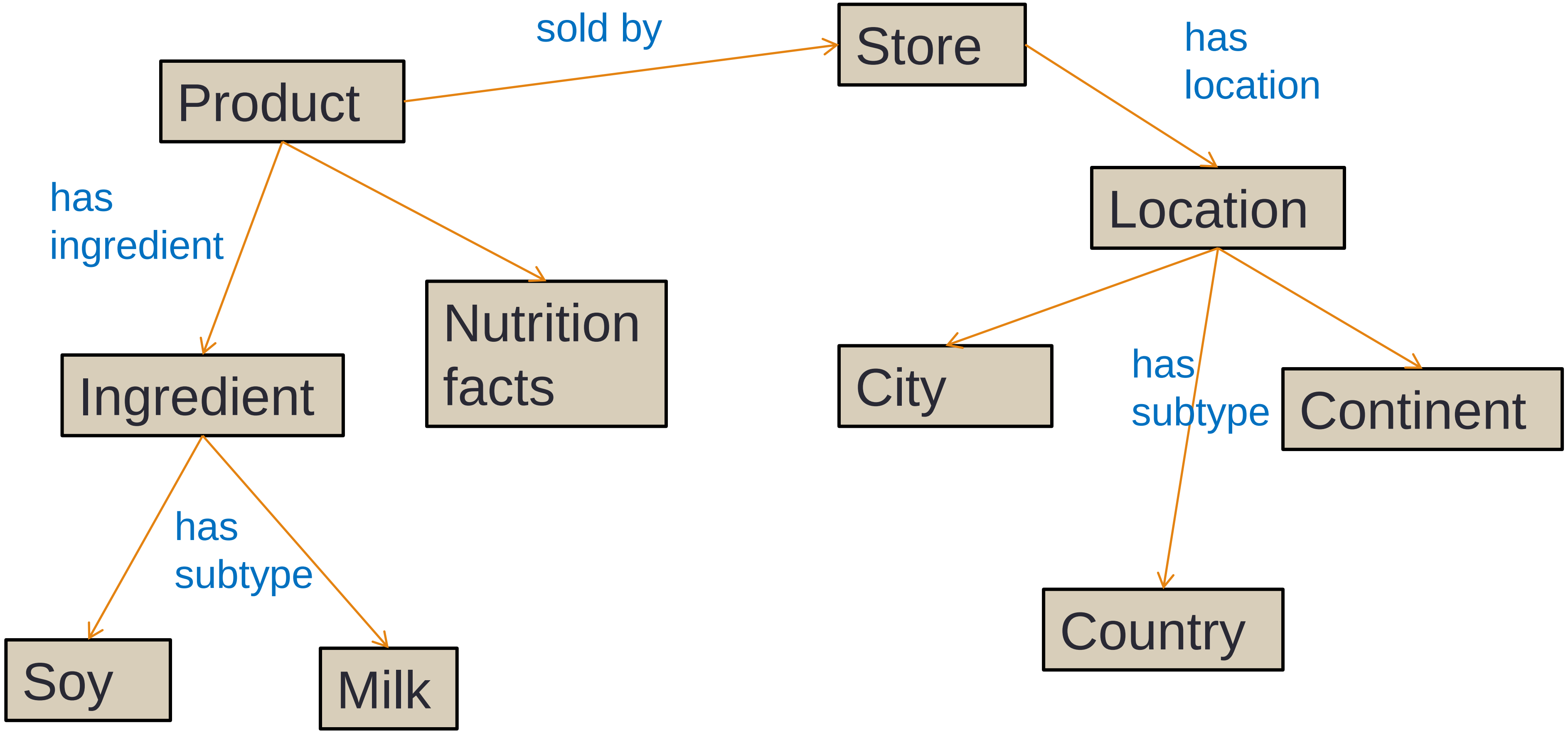
From vocabulary to ontology

- 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

Ontology

Dog SubClassOf Animal

Dalmatian SubClassOf Dog

Patch Type Dalmatian

Pete hasPet Patch

hasPet Domain Person

Example entailments

Dalmatian SubClassOf Animal

Patch Type Dog

hasPet some Dog SubClassOf Person

Pete Type Person

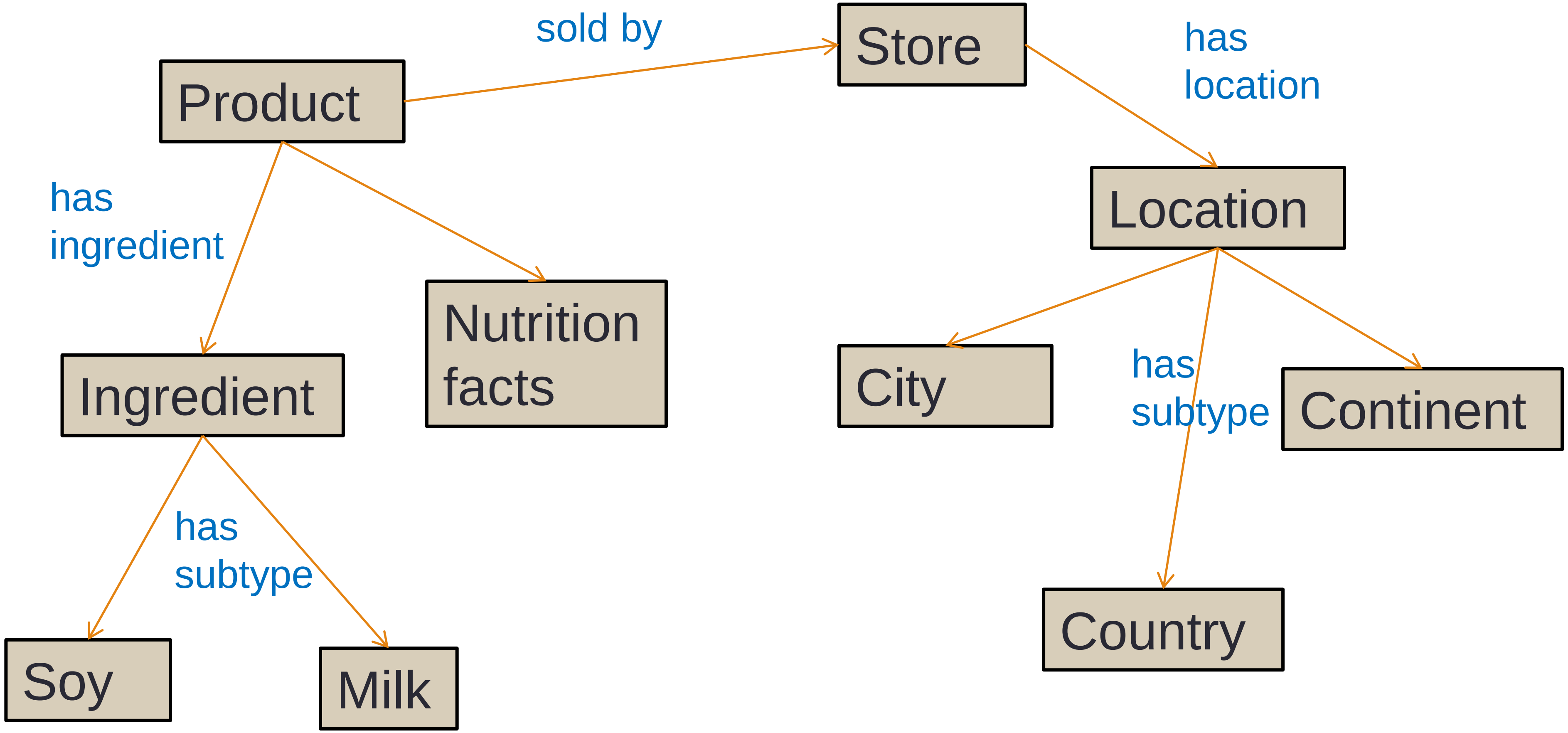
Pete Type hasPet some Dog

Dog SubClassOf Animal

Dalmatian SubClassOf Dog

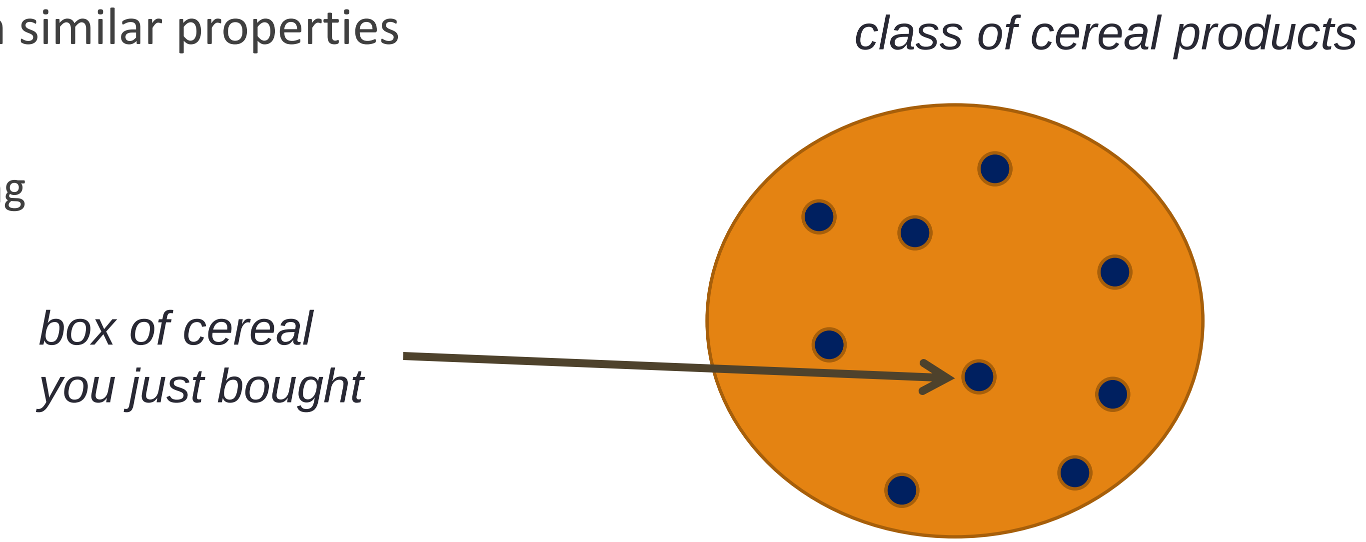
⋮

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

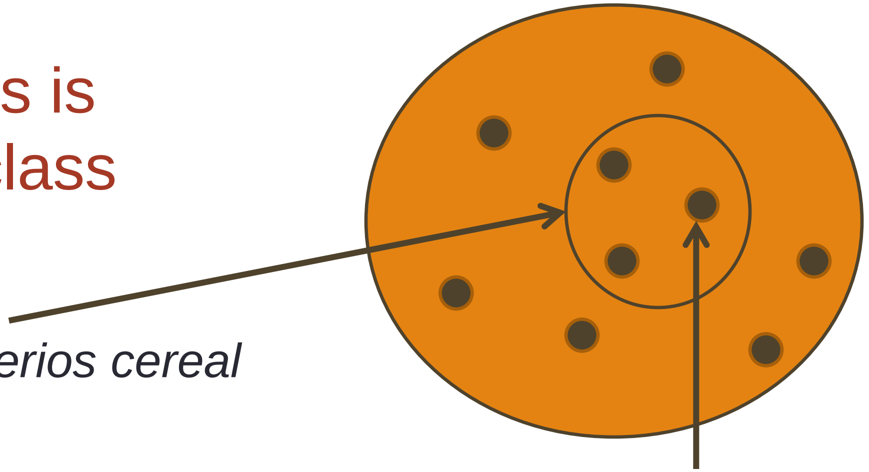


Class inheritance

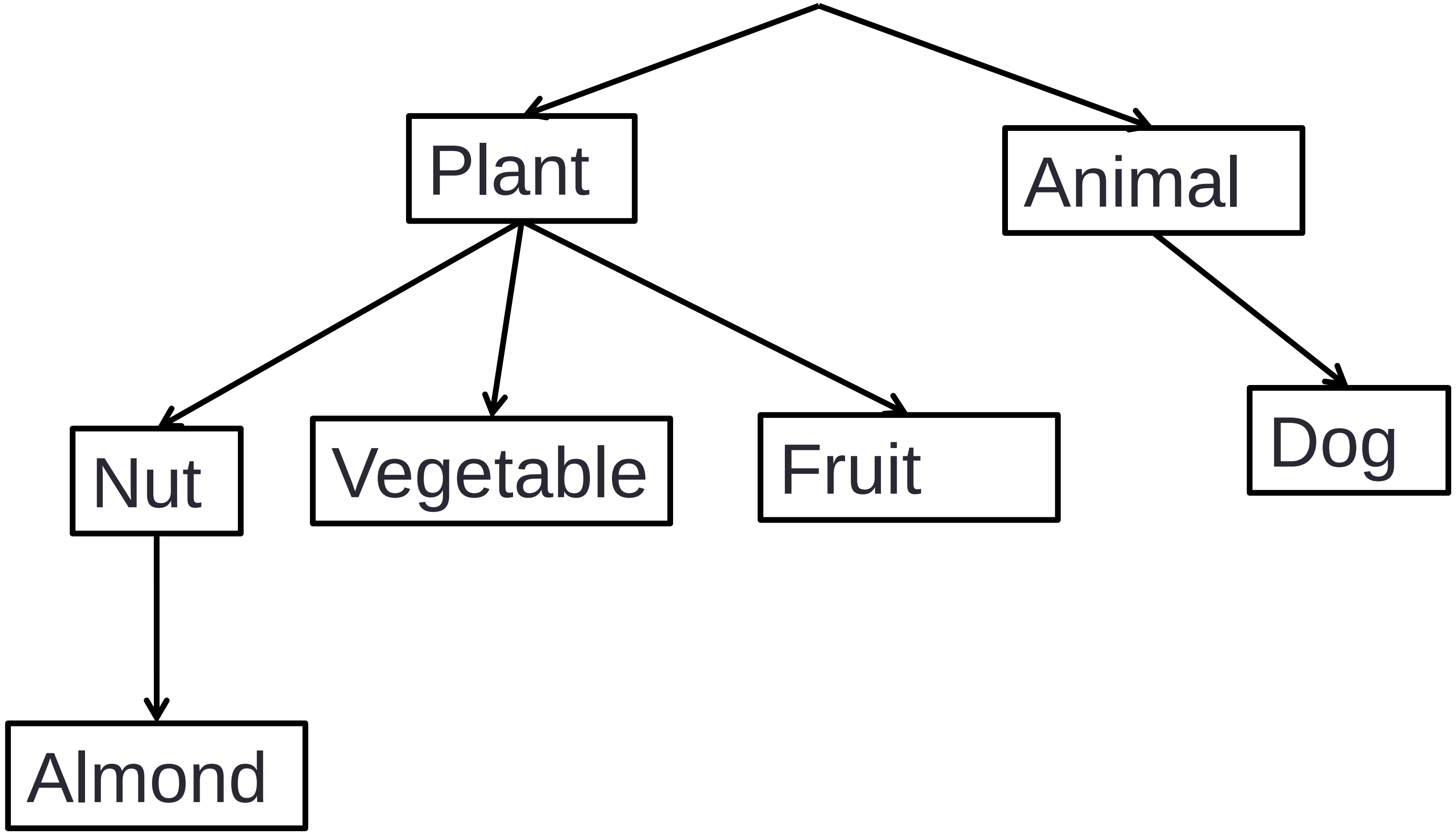
Classes usually constitute a taxonomic hierarchy (a subclass-superclass hierarchy)

an IS-A hierarchy:

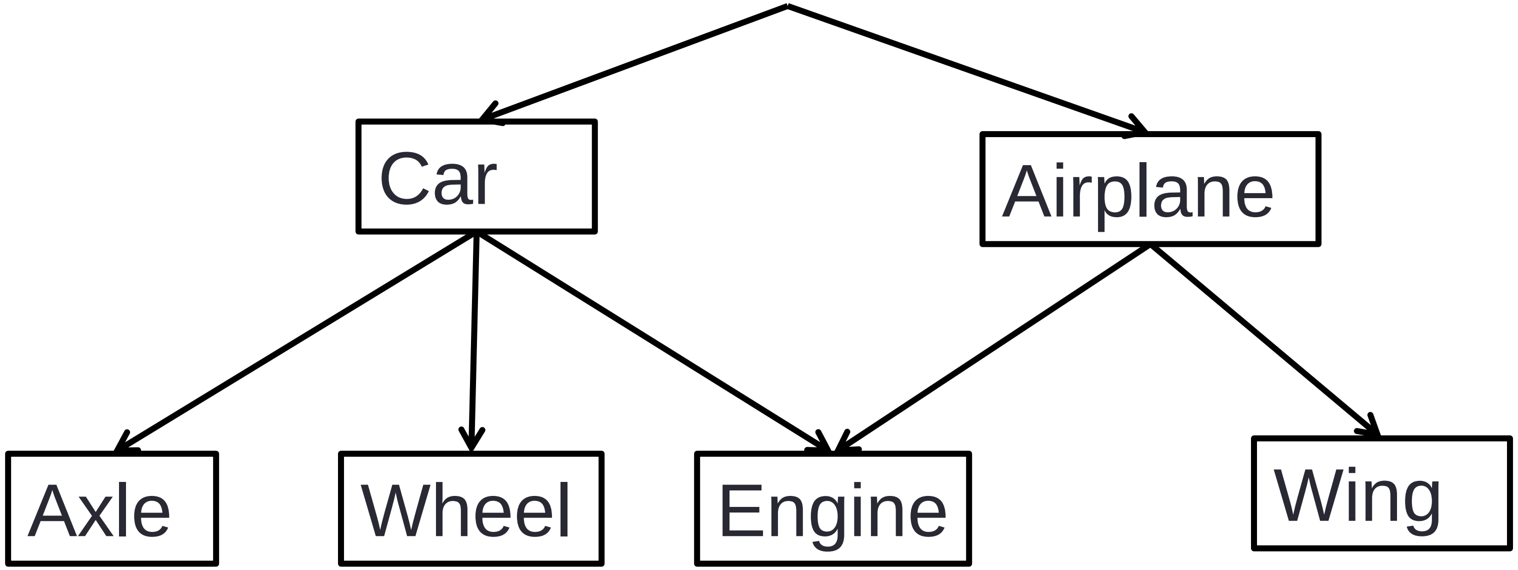
an instance of a subclass is
an instance of its superclass



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



Subclass-Superclass relations?



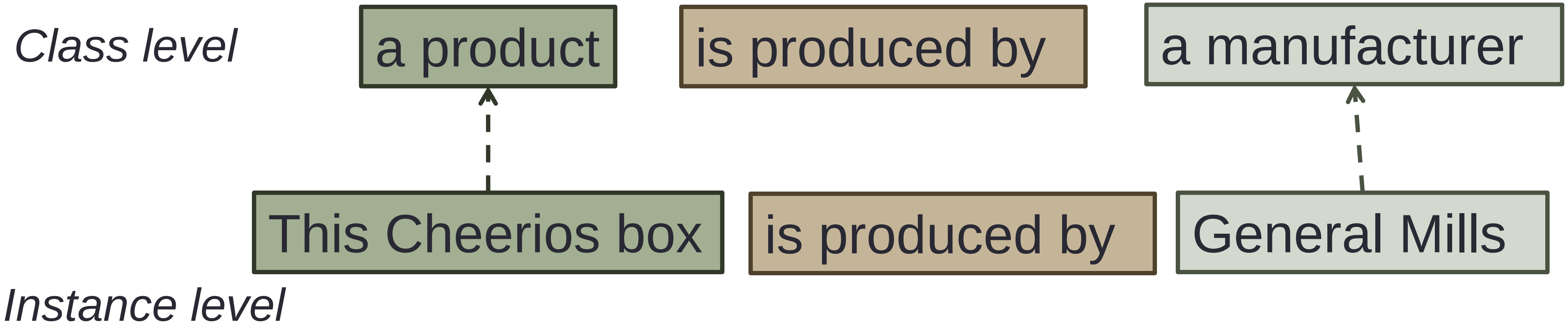
Subclass-Superclass relations?

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

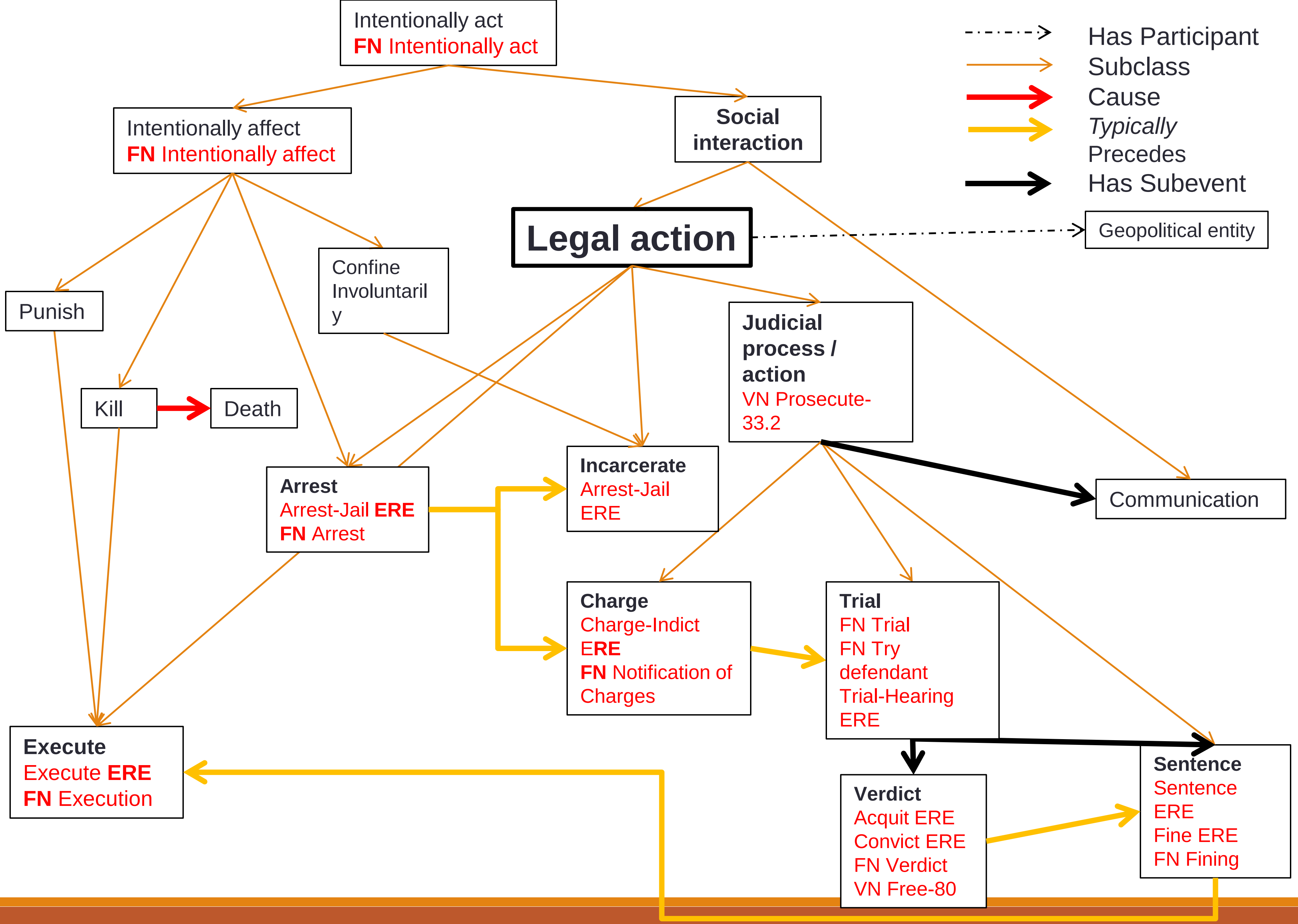
- Individuals are the last level of the ontology; they cannot be further specified
- They represent a materialization of the descriptions at the class level
- This is the level at which the actual data is put in
- The data depends on the application
  - grocery app?
  - tracking terrorist organizations?

Ontologies for NLP

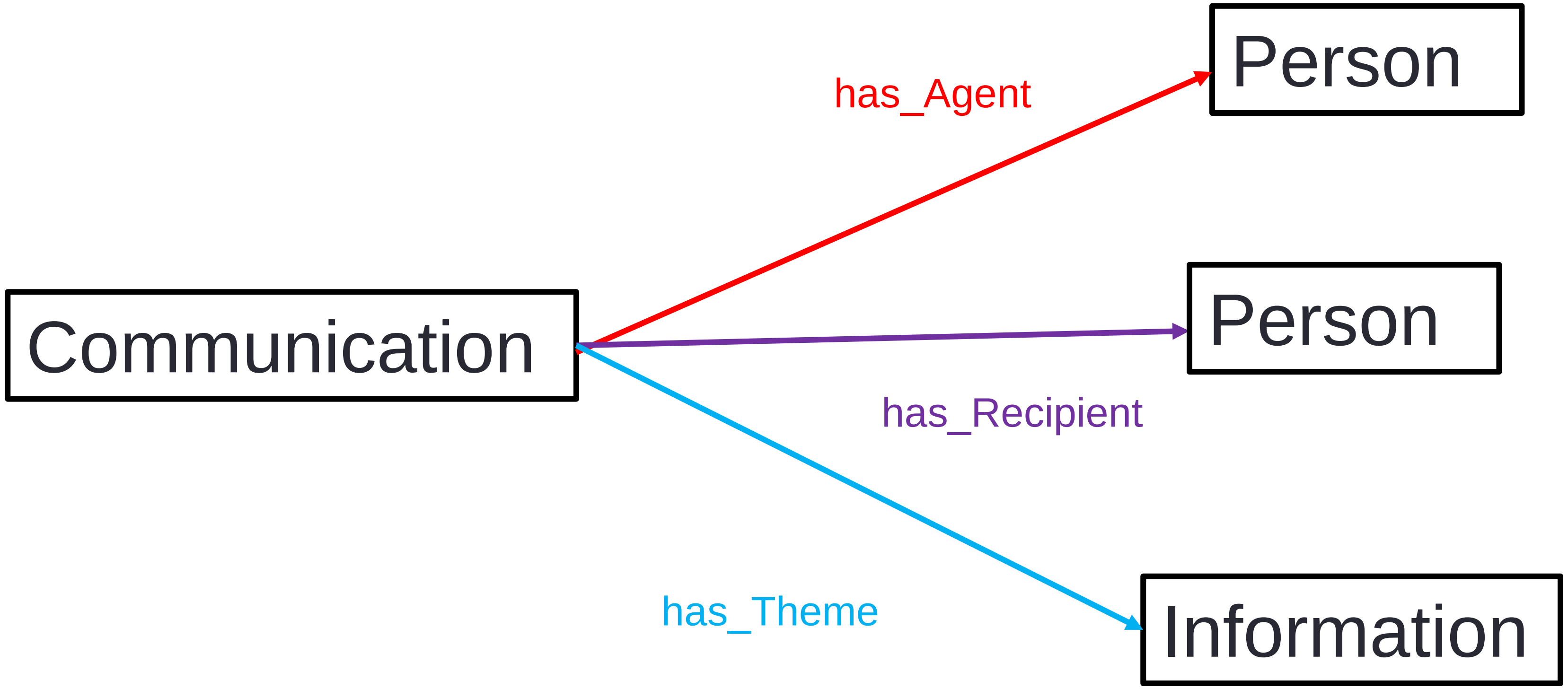
- Move from words to concepts/entities underlying the words
- A conceptual ontology with links to lexical items
- Bio-NLP
- Event extraction and participant tracking

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)


Intentionally act
FN Intentionally act
Intentionally affect
FN Intentionally affect
Social interaction
Has Participant
Subclass
Cause
Typically Precedes
Has Subevent
Legal action
Geopolitical entity
Confine Involuntaril y
Punish
Judicial process / action
VN Prosecute-33.2
Kill
Death
Arrest
Arrest-Jail ERE
FN Arrest
Incarcerate
Arrest-Jail ERE
Communication
Charge
Charge-Indict ERE
FN Notification of Charges
Trial
FN Trial
FN Try defendant
Trial-Hearing ERE
Execute
Execute ERE
FN Execution
Verdict
Acquit ERE
Convict ERE
FN Verdict
VN Free-80
Sentence
Sentence ERE
Fine ERE
FN Fining

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

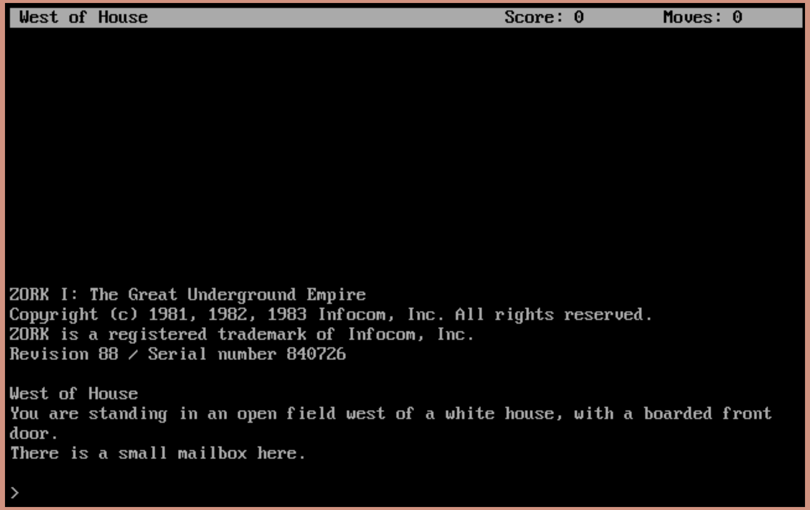
Temporal	
Start and End	(times specific to event)
Duration	1 minute through multiple years

What is the problem with this approach?

- What might happen when using an ontology like this in an actual application?
  - The system might have problems staying at the appropriate level
  - Are all entities, events, properties captured? And if it's too big, can it be processed? (scalable)
  - How do events interact?

The Story So Far… (Recap)

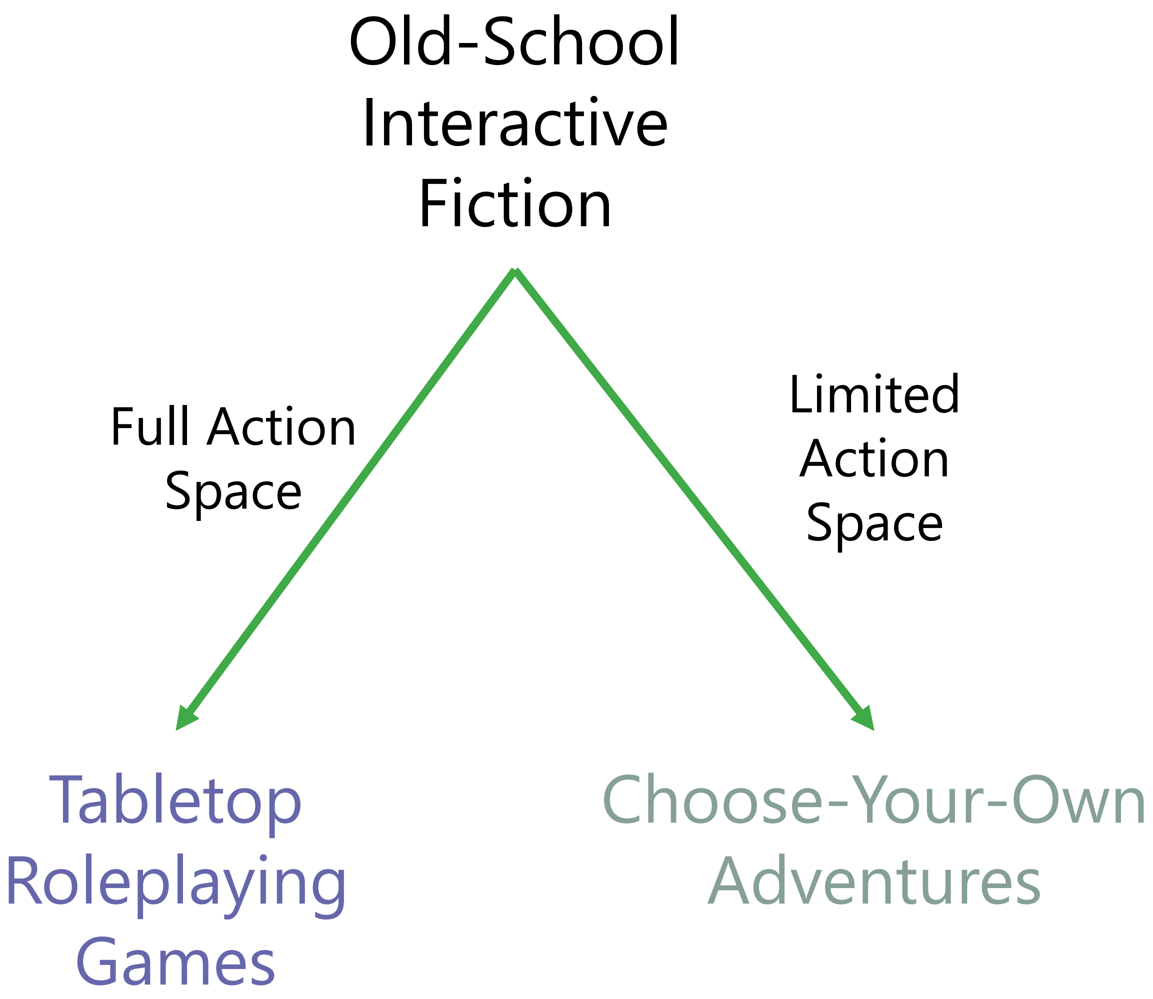
Interactive Fiction & Storytelling



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”?

Cohesion & coherence
Logical flow, no plot holes or loose ends, fluency
Consistency with story world
Compelling/dynamic narrative, evoke emotions
Character development, relatable characters
Detailed world
Consequences of actions/events
Implied lesson
Good use of medium

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

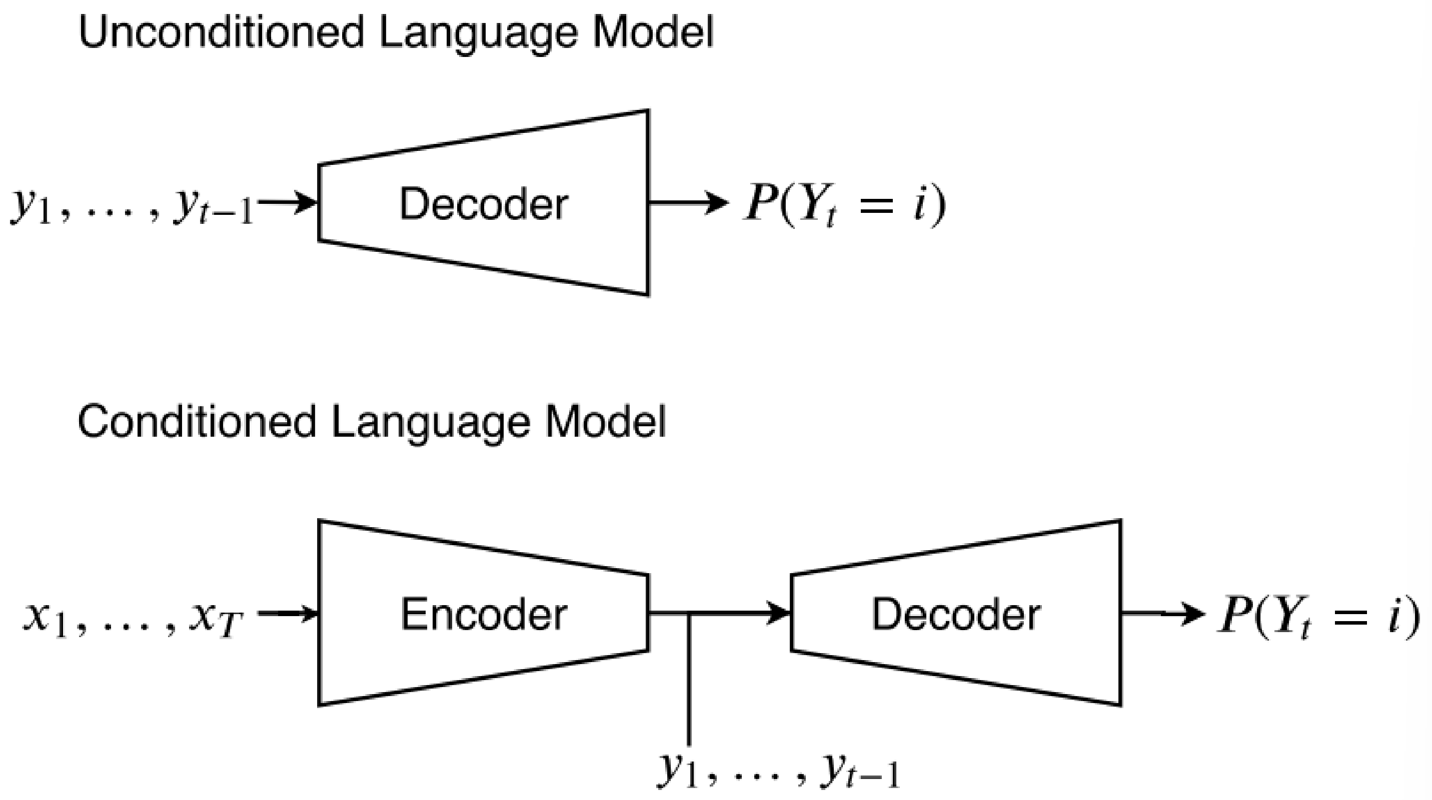
Complexity/Theme

- Underlying ideas/themes
- Multiple plot elements
- underlying deep / philosophical themes
- Satisfying to read, gives interesting insights

NEURAL SYSTEMS

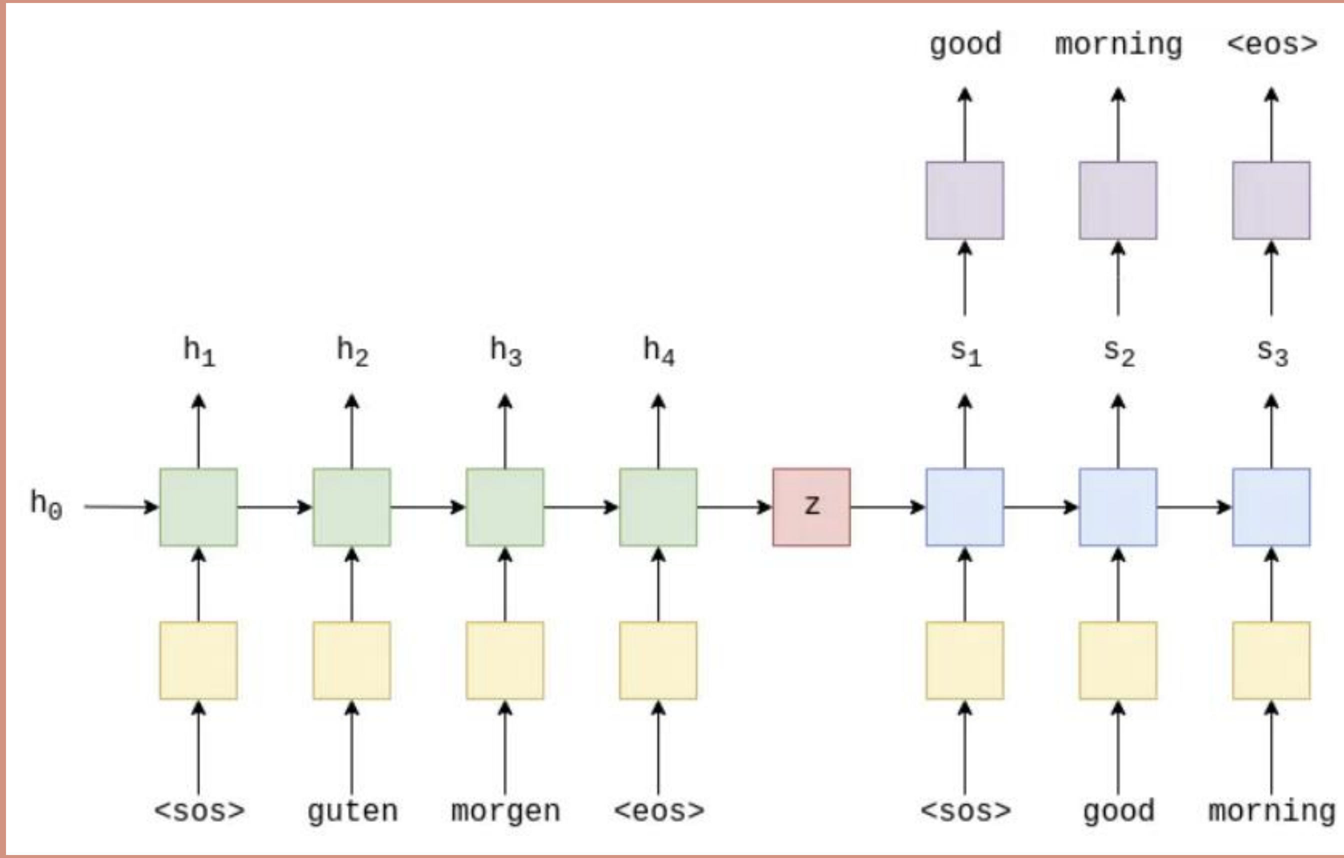
Neural Generation

- Probabilistic
  - Unconditioned
    $P(Y)$
  - Conditioned
    $P(Y \mid X)$



From Daphne Ippolito's lecture

RNNs (Sequence-to-Sequence)



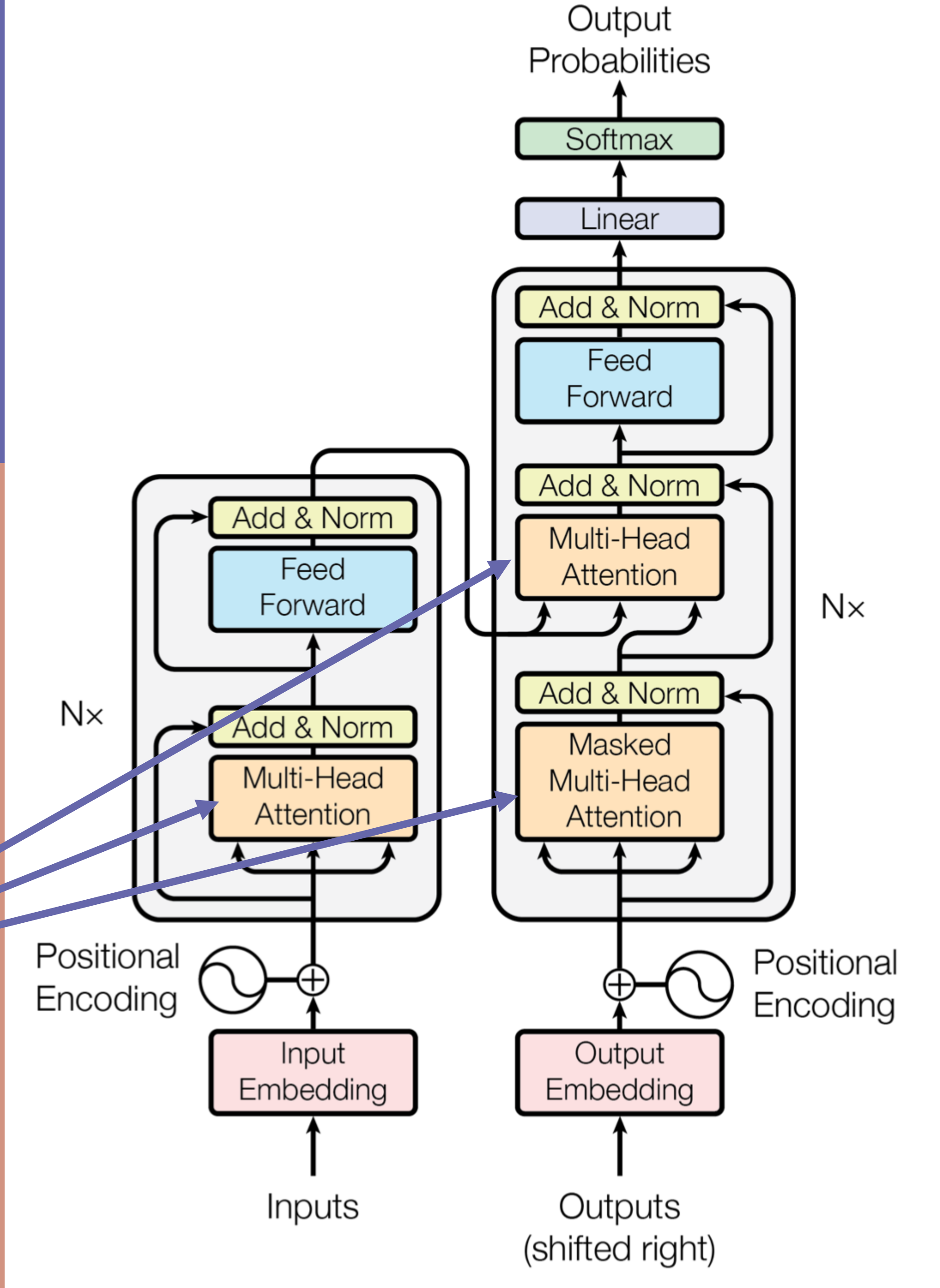
https://www.jianshu.com/p/5b06a91d4f3d

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!

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

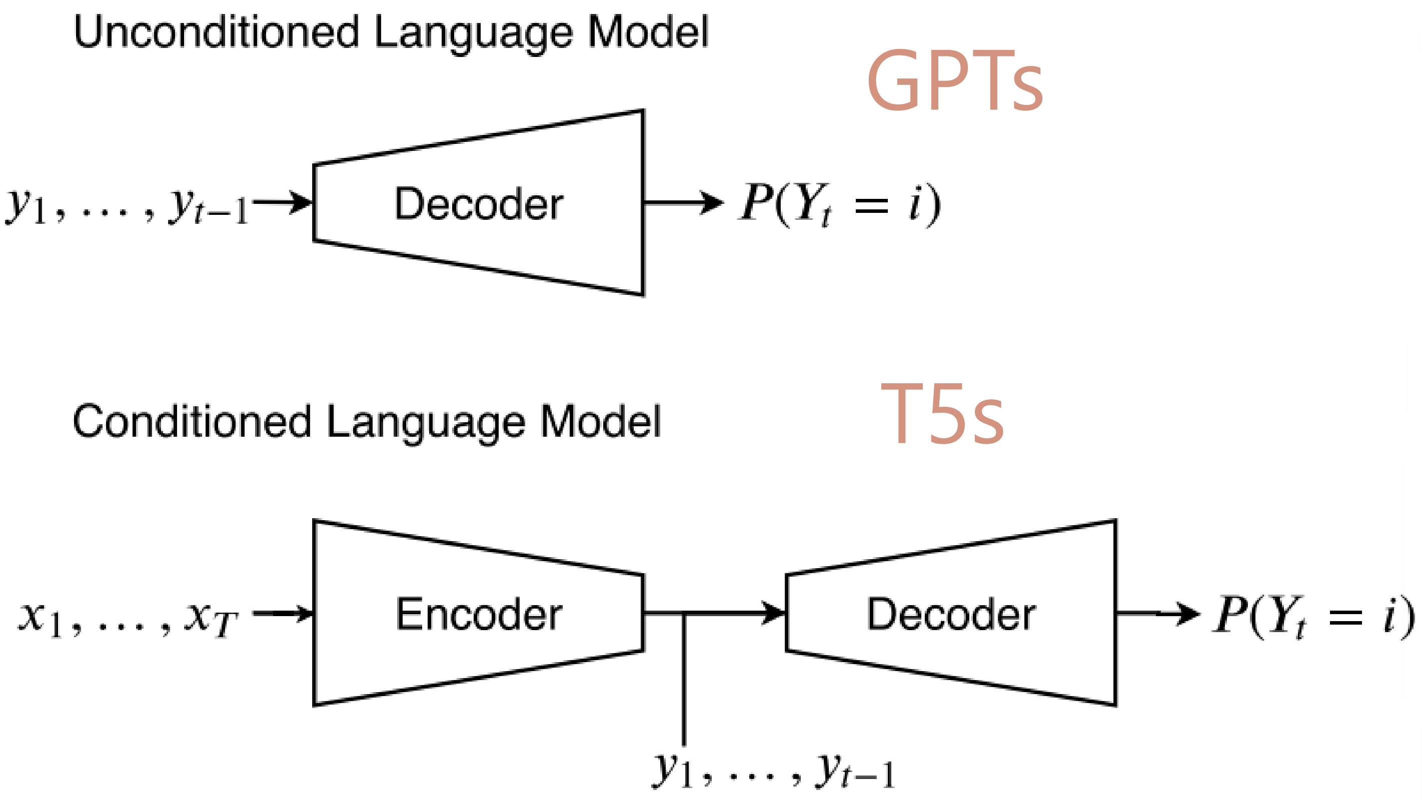


From "Attention is All You Need"

Transformer Types

Encoder-Only: BERTs

What are encoder-only models useful for?



From Daphne Ippolito's lecture

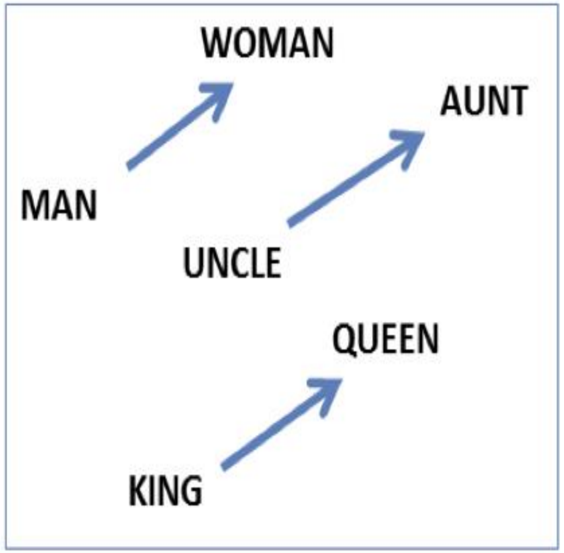
(Some) Properties of Embeddings

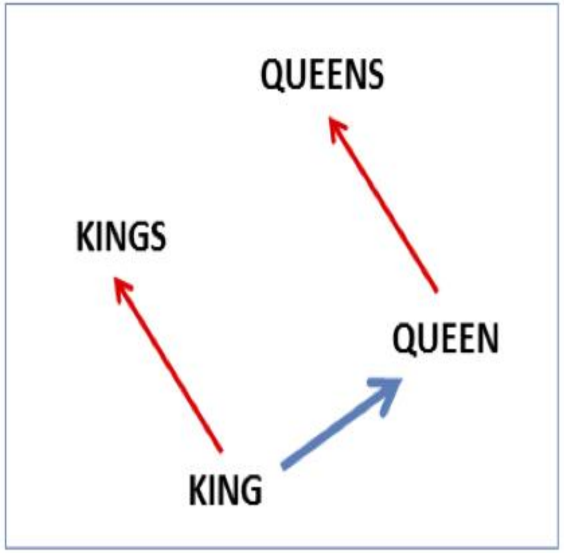
Why are embeddings useful for neural networks?

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



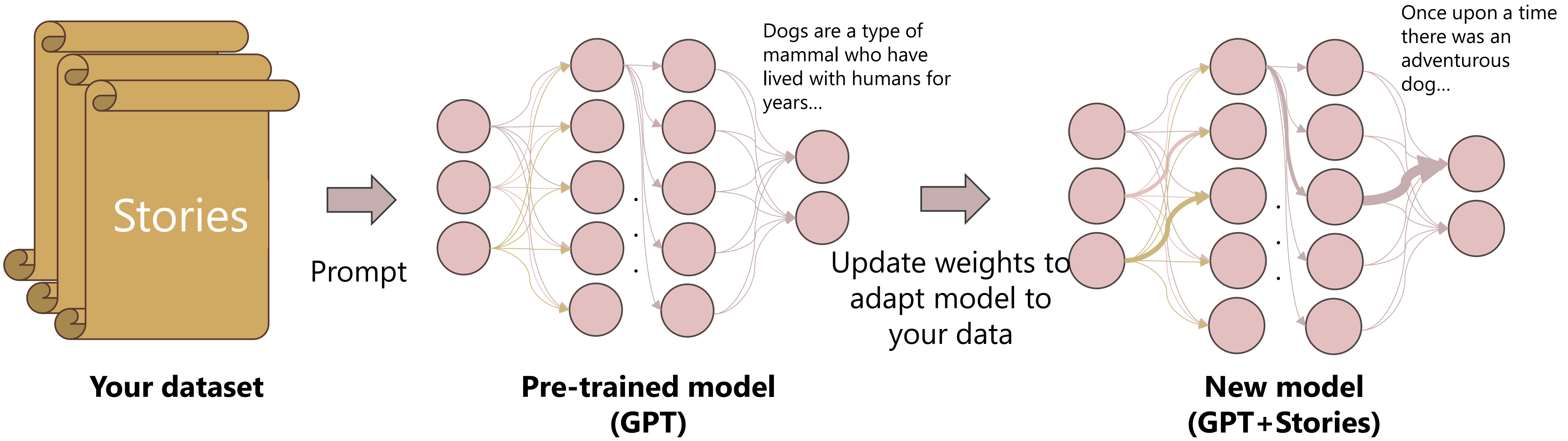


vector(*'king'*) –
vector(*'man'*) +
vector(*'woman'*) ≈
vector('queen')

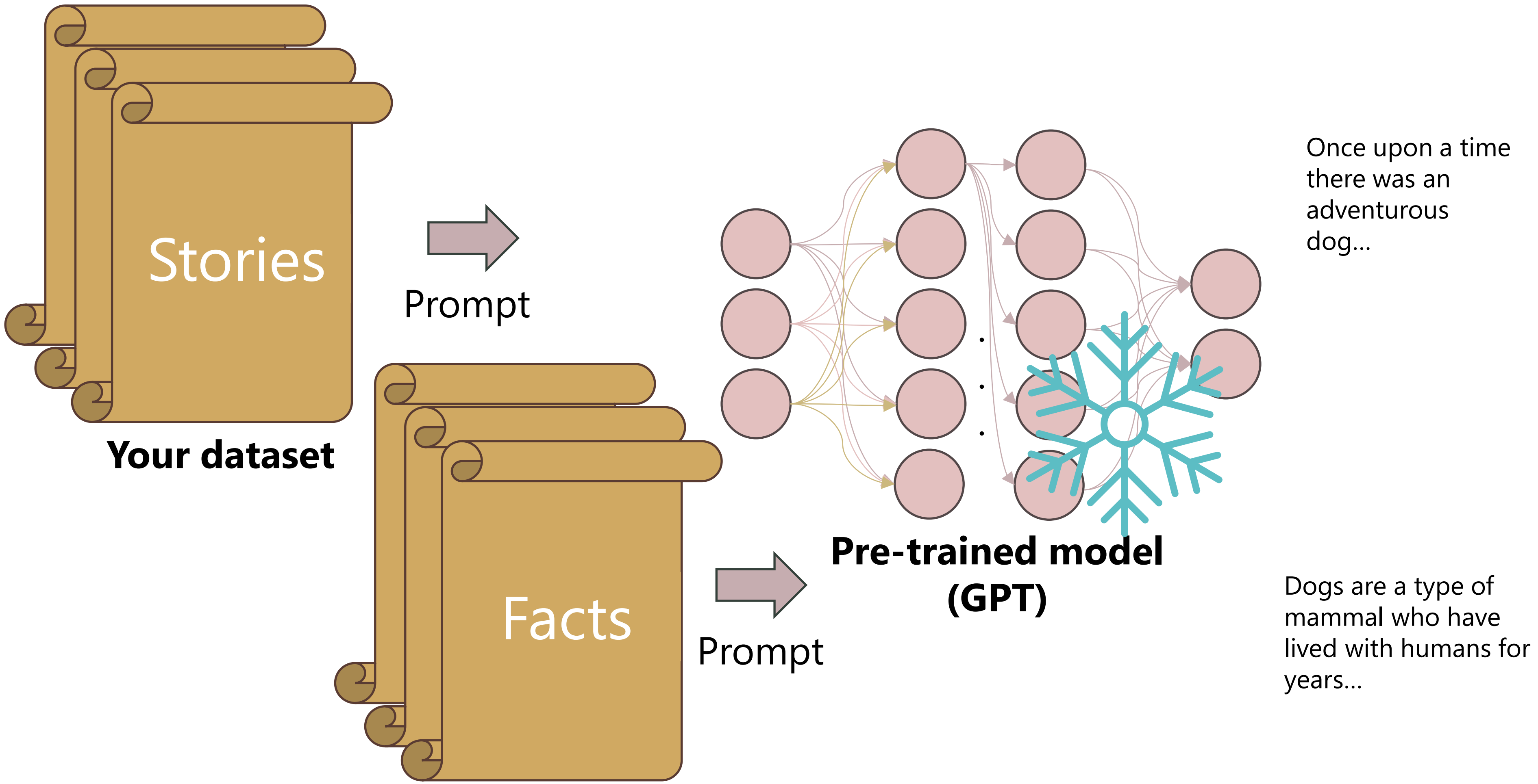
vector(*'Paris'*) -
vector(*'France'*) +
vector(*'Italy'*) ≈
vector('Rome')

Mikolov et al. (2013)

Finetuning


Stories
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
Your dataset
Prompt
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.

Instructions

Task

Sentence:
The dog ate the giant fish.

Model

Output

Few-shot Prompting

Instructions

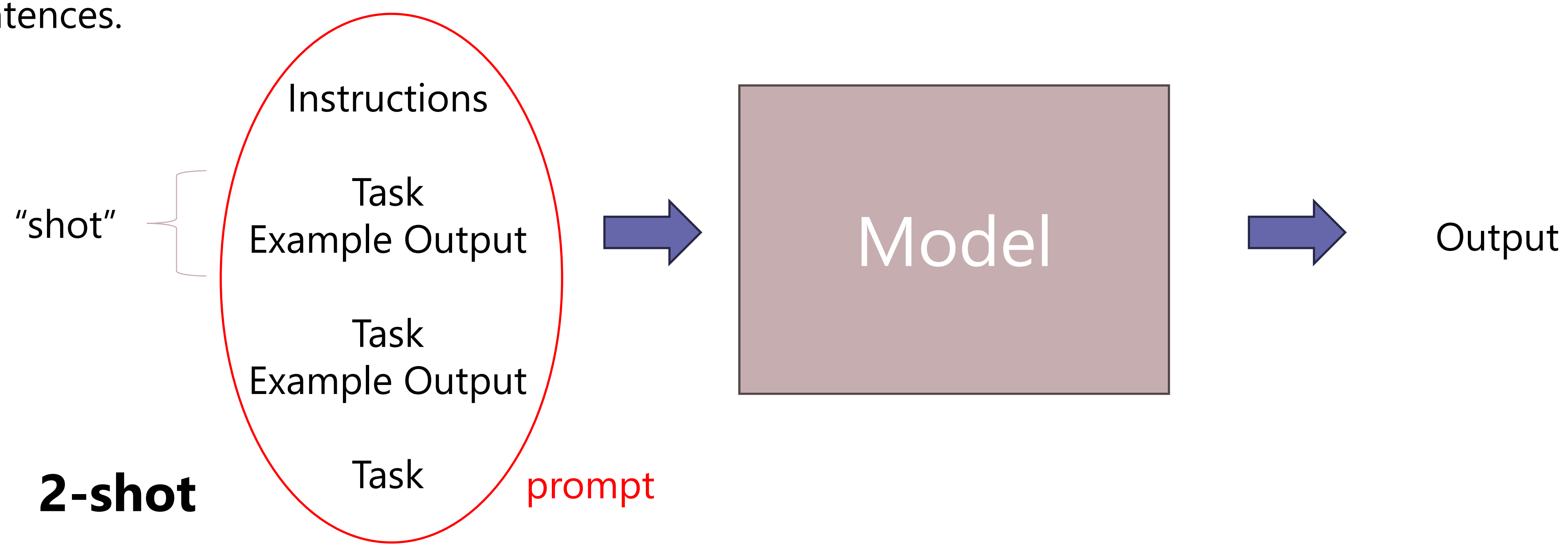
You are a helpful assistant. You will be tagging the parts of speech in sentences.

Task

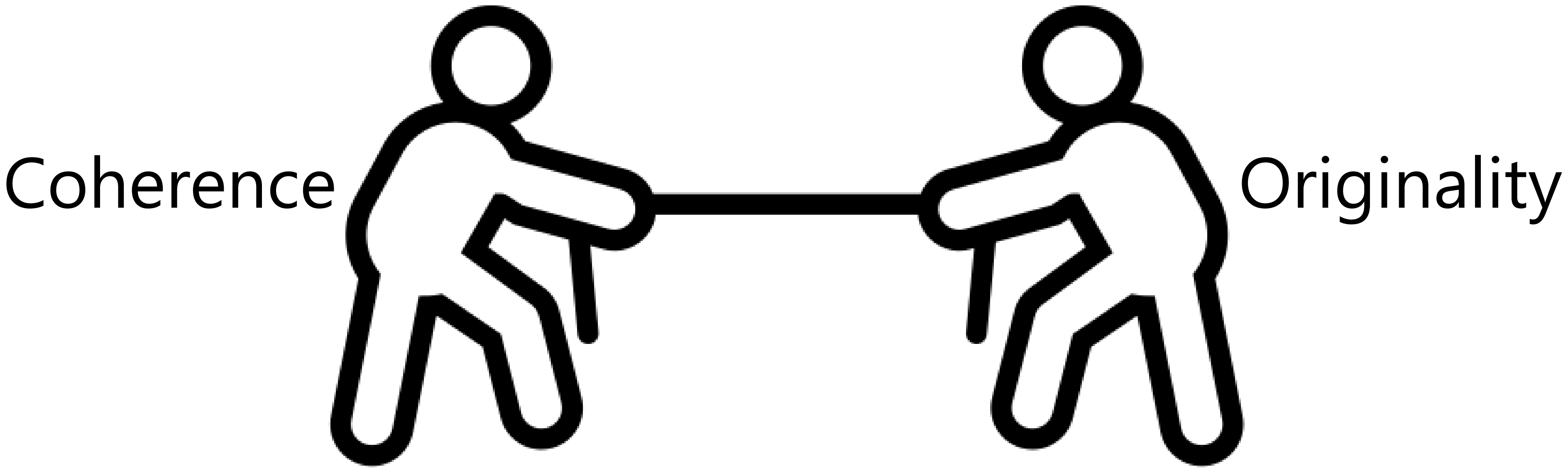
Sentence:
The dog ate the giant fish.

Example Output

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



Lara's Language Model Tradeoff



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

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."
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.
"Thank you," he said. "I'm glad you're my dog."
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."
She nodded and lay down on the floor.
Her owner got up, turned off the lights, and lay down on the bed.

On Theme ✓

Grammar ✓

Remembering Story State ✗

Commonsense Reasoning ✗

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.

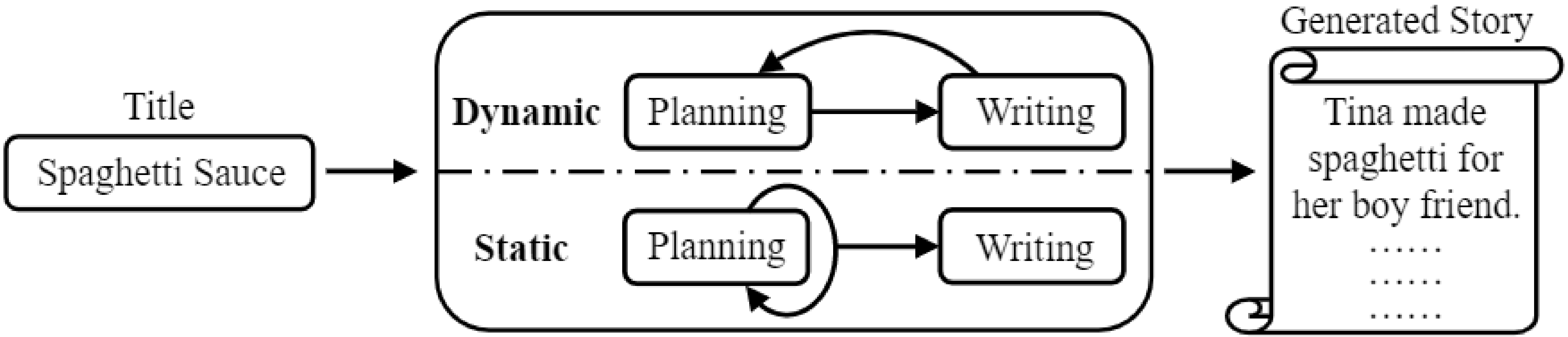
Learning Objectives

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



Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., & Yan, R. (2019). Plan-And-Write: Towards Better Automatic Storytelling. *AAAI Conference on Artificial Intelligence (AAAI), 33*(1), 7378–7385. https://aaai.org/ojs/index.php/AAAI/article/view/4726

Re3

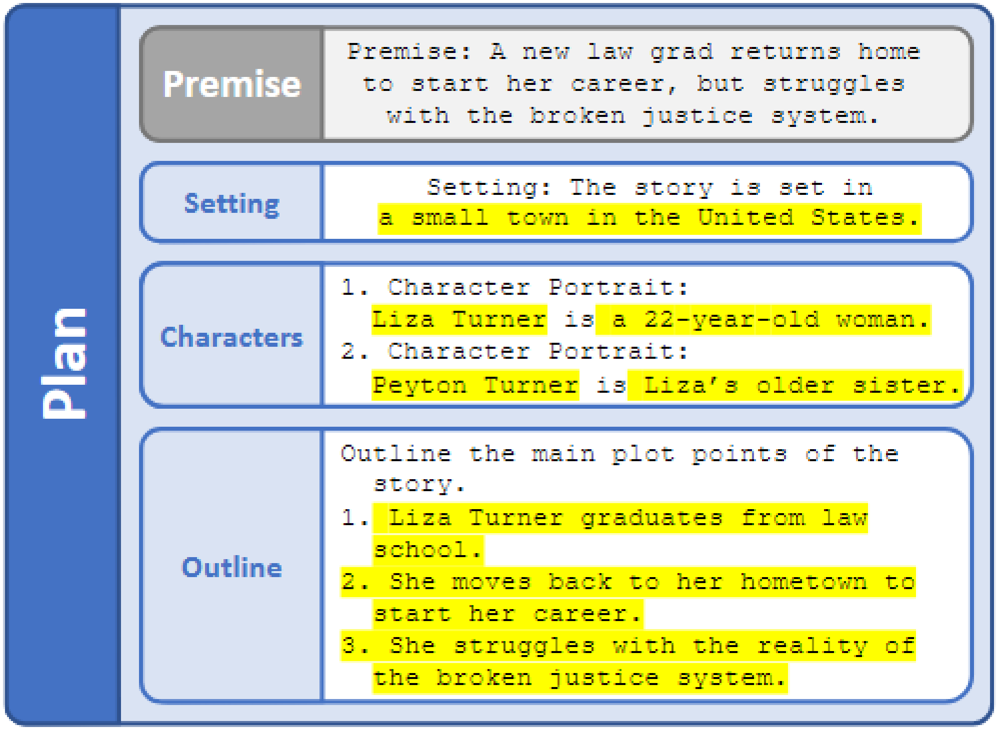


Figure 2: Illustration of Re^3's Plan module, which prompts a language model to generate a setting, characters, and outline based on the premise. Highlighting indicates generated text.

Re^3: Generating Longer Stories With Recursive Reprompting and Revision

Kevin Yang[1] **Yuandong Tian**[2] **Nanyun Peng**[3] **Dan Klein**[1]
[1]UC Berkeley, [2]Meta AI, [3]UCLA
{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 (Re^3) 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 Re^3's stories as having a coherent overarching plot (by 14% absolute increase), and relevant to the given initial premise (by 20%).

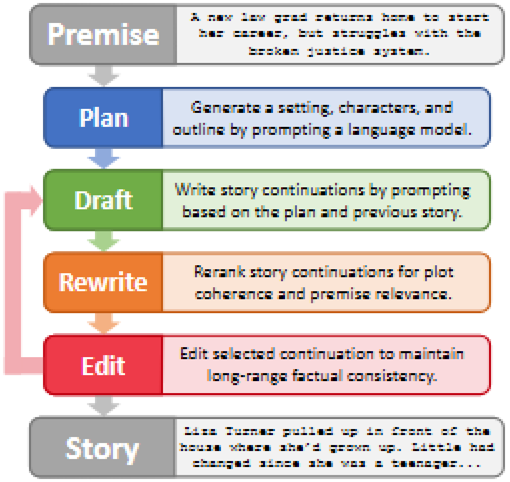


Figure 1: High-level overview of Re^3.

1 Introduction

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

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

K. Yang, Y. Tian, N. Peng, and D. Klein, "Re^3: Generating Longer Stories With Recursive Reprompting and Revision," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4393–4479. doi: 10.18653/v1/2022.emnlp-main.296.

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:

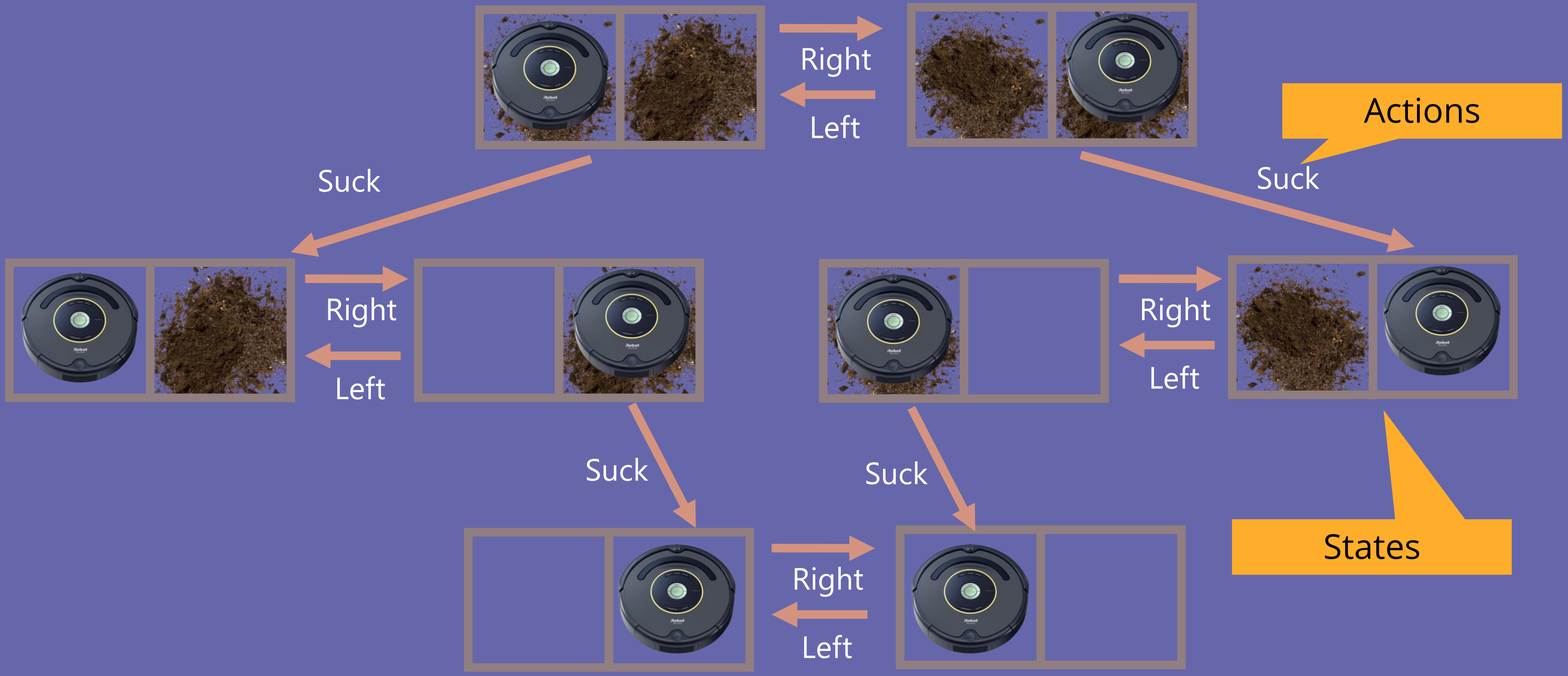
- Prompt an LLM to generate stories given varying amounts of context
- Implement NLP evaluation metrics using existing libraries
- Compare the quality of guided vs unguided story generation
- Determine the adequacy of automated metrics like BLEU and ROGUE for creative evaluation

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


Right
Left
Actions
Suck
Suck
Right
Left
Right
Left
States
Suck
Suck
Right
Left

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

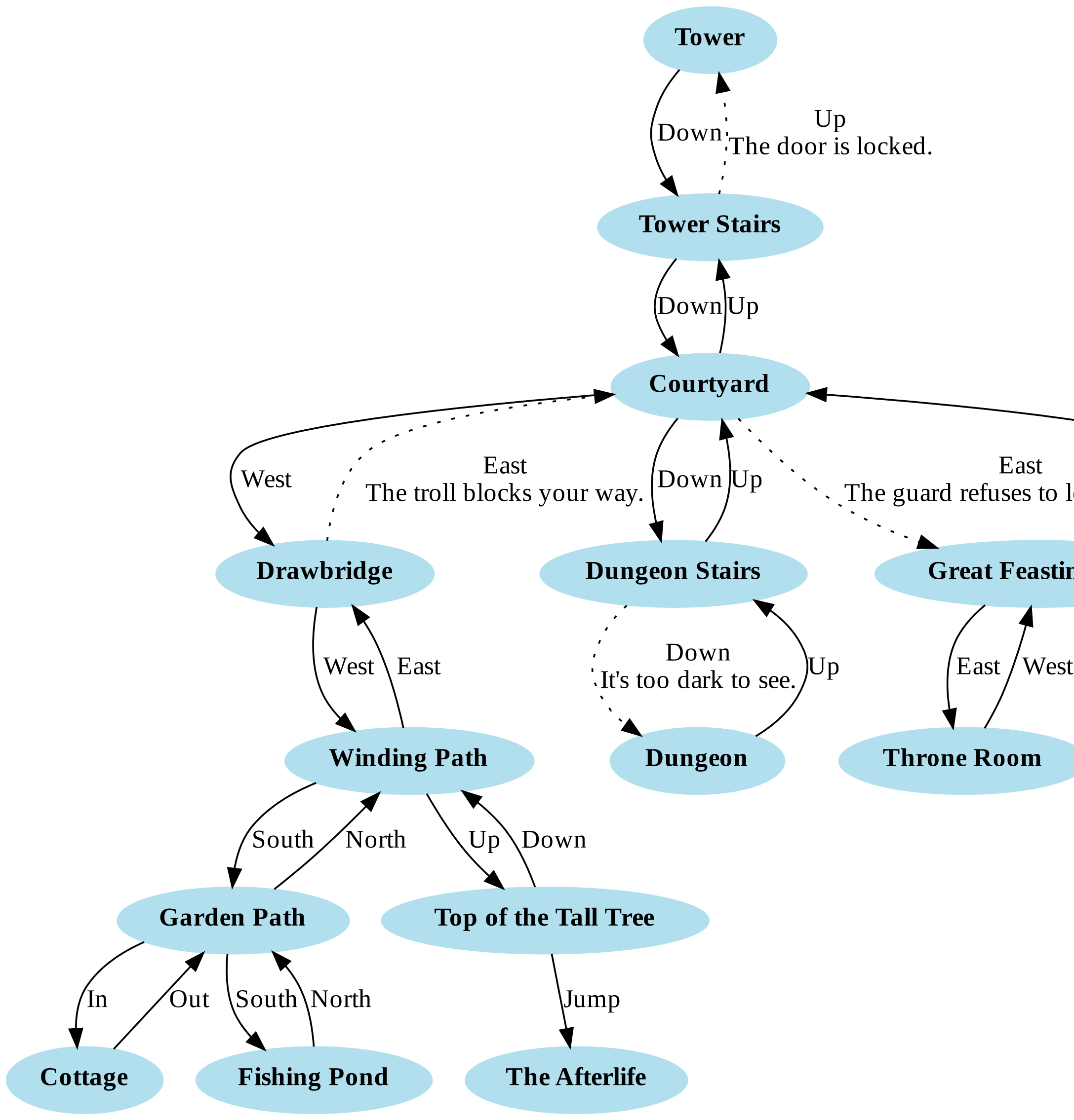
Actions: move between connected locations

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 sate 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)

L. J. Martin, "Neurosymbolic Automated Story Generation," PhD, Georgia Institute of Technology, Atlanta, GA, 2021. https://smartech.gatech.edu/handle/1853/64643

Pre-Conditions and Effects

Jen sent the book to Remy from Baltimore.

Pre-Conditions → Event → Effects

Pre-Conditions

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

Effects

!has_location(book, Baltimore)
has_location(book, Remy)

L. J. Martin, "Neurosymbolic Automated Story Generation," PhD, Georgia Institute of Technology, Atlanta, GA, 2021. https://smartech.gatech.edu/handle/1853/64643

What are we planning over?

Structure (Schemas)

KB Schemas

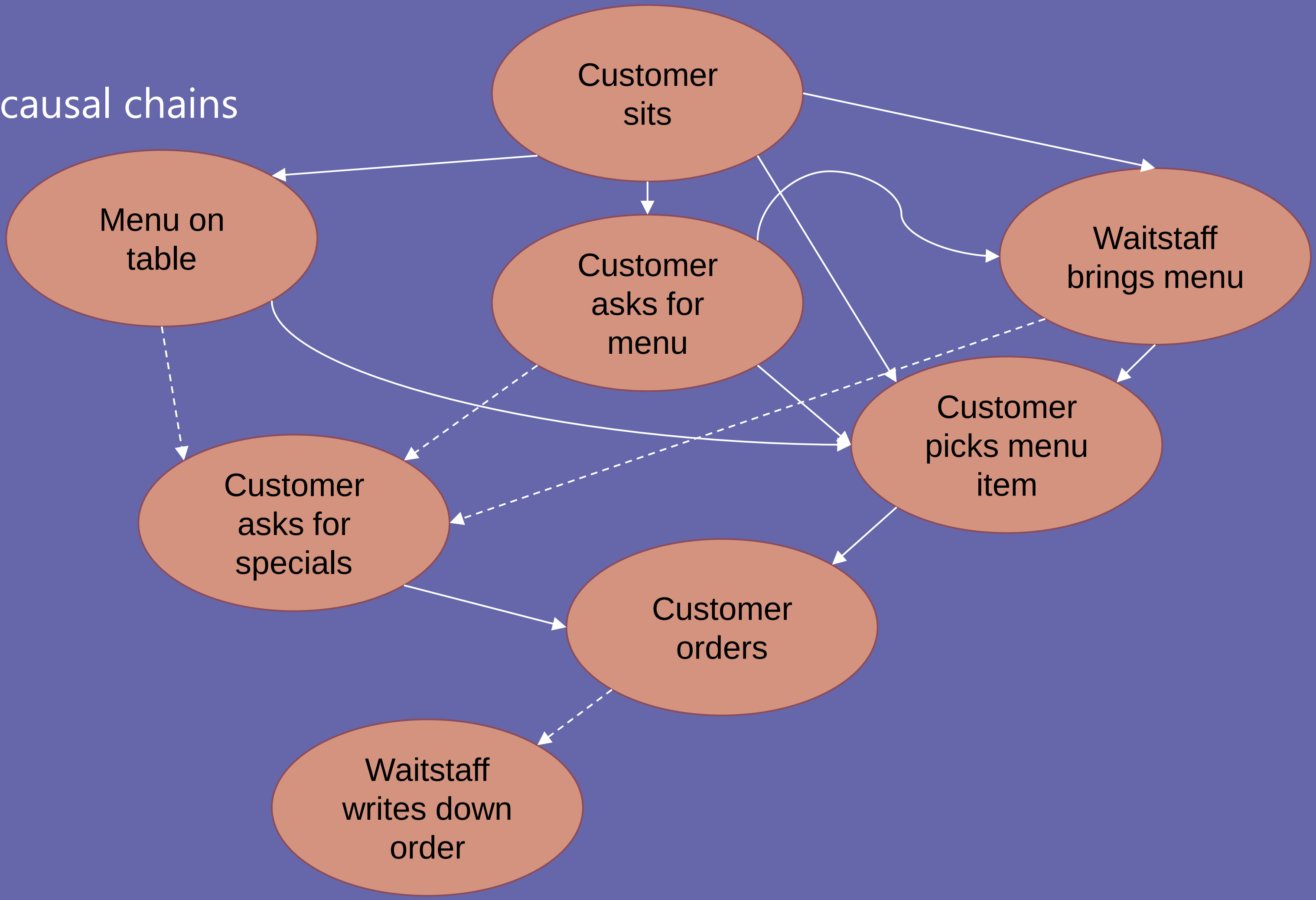
Scripts

Procedures

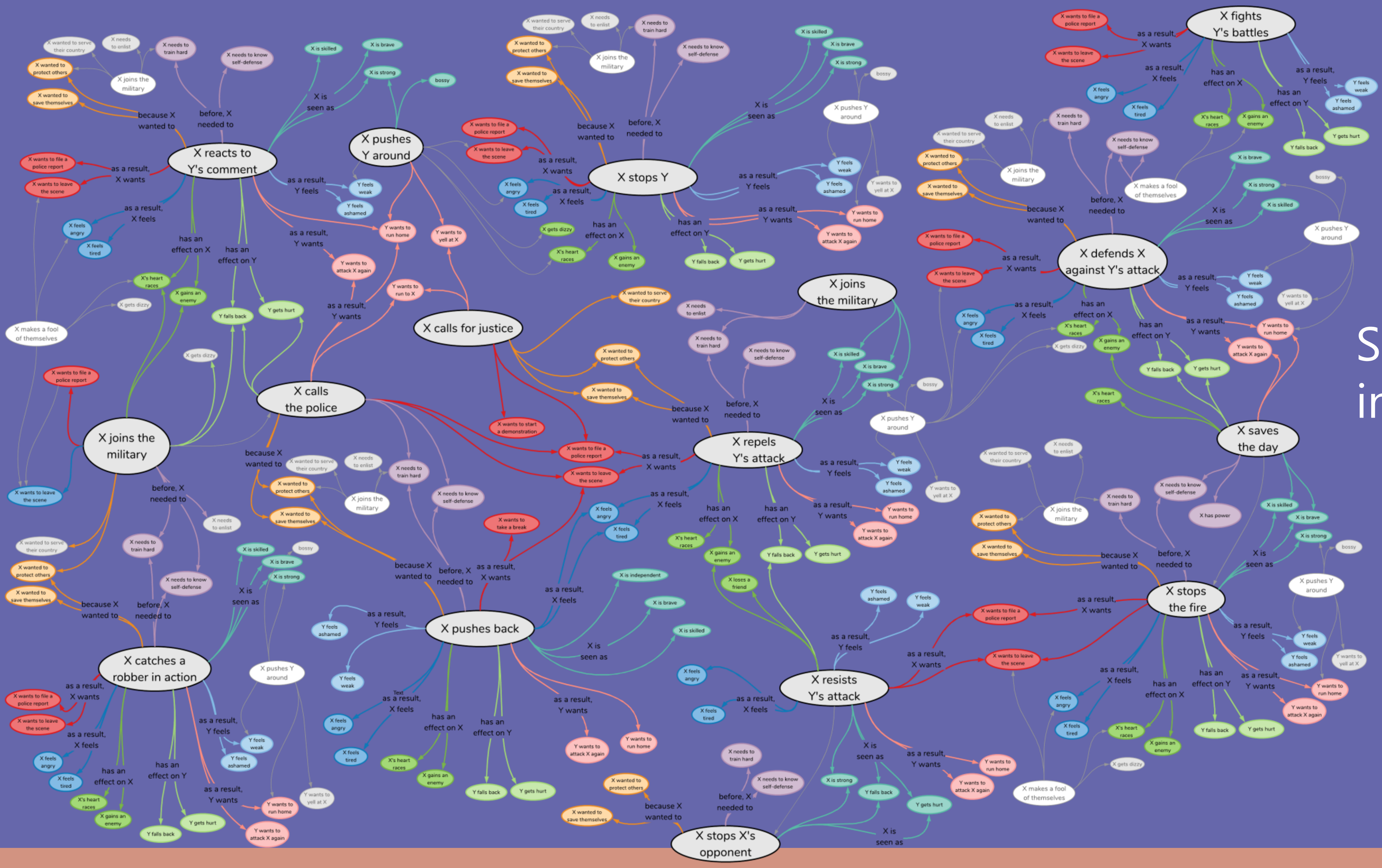
Organization of Commonsense Knowledge

Scripts

Common sequence of causal chains



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



Sequence vs structured information

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

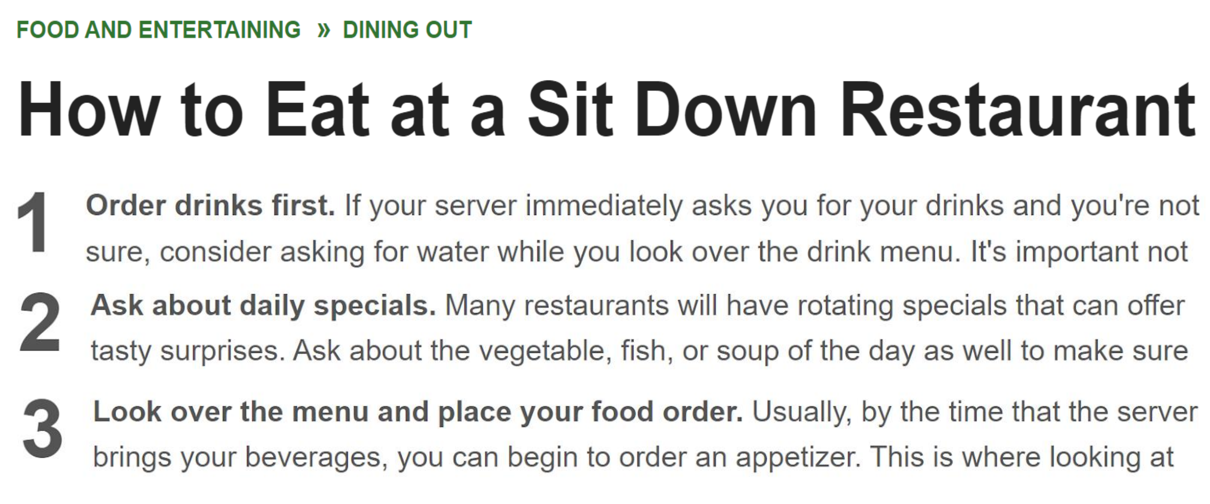
Martin, Lara J., et al. "Event representations for automated story generation with deep neural nets.“ *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

Procedures: Script with a goal

category

goal

steps



FOOD AND ENTERTAINING » DINING OUT

How to Eat at a Sit Down Restaurant

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

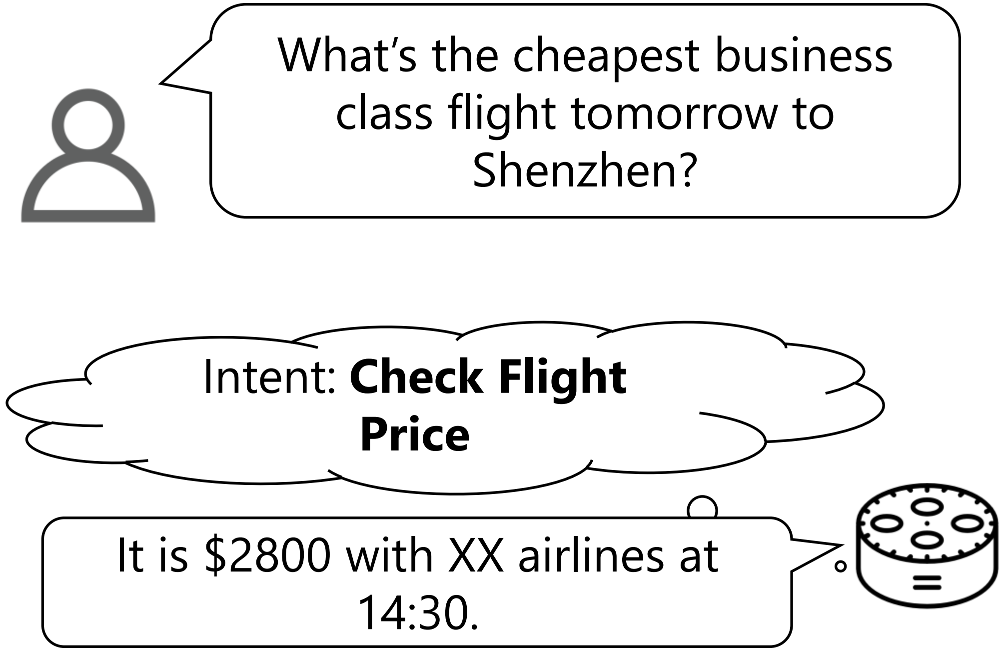
2 **Ask about daily specials.** Many restaurants will have rotating specials that can offer tasty surprises. Ask about the vegetable, fish, or soup of the day as well to make sure

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

From Veronica Qing Lyu & Harry Li Zhang's lecture

Intent Detection

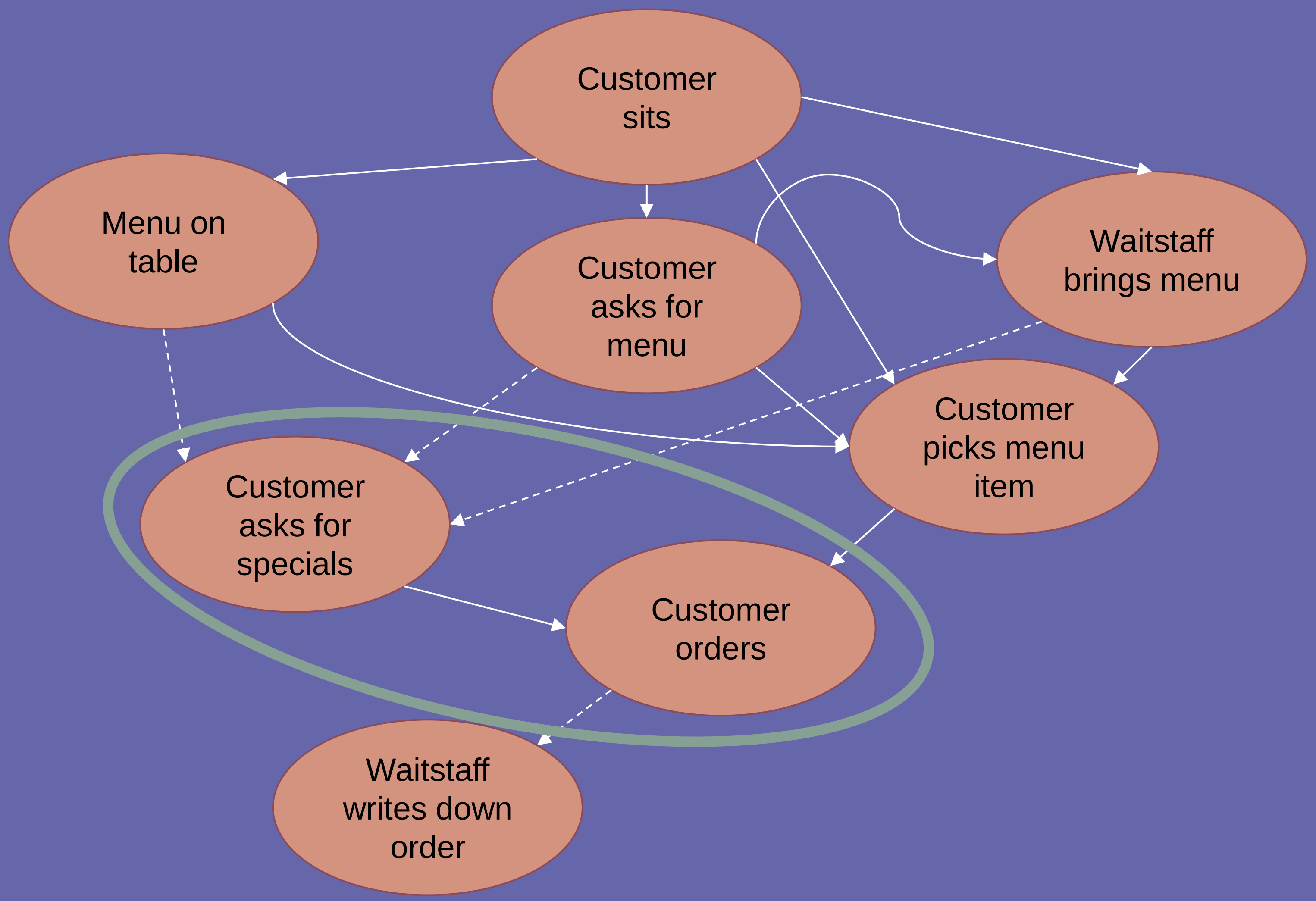
- Task-oriented dialog systems needs to match an **utterance** to an **intent**, before making informed responses
- Sentence classification task
  - Given an utterance, and some candidate intents
  - Choose the correct intent



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**

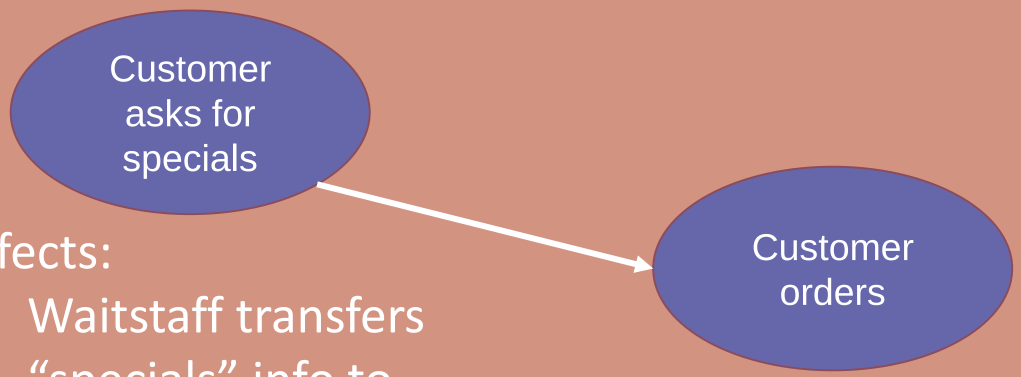
(Zhang et al., 2020): Intent Detection with WikiHow

Scripts



Customer sits
Menu on table
Customer asks for menu
Waitstaff brings menu
Customer picks menu item
Customer asks for specials
Customer orders
Waitstaff writes down order

Causal Links



Effects:
- Waitstaff transfers “specials” info to customer
- Customer knows what they want to order

Precondition:
- Customer knows what they want to order

Causal Links→ Actions for Planning

$$\boldsymbol{a}\text{:}\quad \boldsymbol{buy(Tom, Potion, Merchant, Market)}$$

$$\text{PRE}(a)\text{:}\quad at(Tom) = Market \wedge at(Merchant) = Market \wedge$$
$$at(Potion) = Merchant \wedge wealth(Tom) \geq 1$$

$$\text{EFF}(a)\text{:}\quad at(Potion) = Tom \wedge wealth(Merchant) \mathrel{+}= 1 \wedge$$
$$wealth(Tom) \mathrel{-}= 1$$

From Stephen Ware's lecture

Representation Language

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

Action name

Variables

```
(:action go
  :parameters (?dir - direction
               ?p - player
               ?l1 - location ?l2 - location)
  :precondition (and
               (at ?p ?l1)
               (connected ?l1 ?dir ?l2)
               (not (blocked ?l1 ?dir ?l2)))
  :effect (and
               (at ?p ?l2)
               (not (at ?p ?l1)))
)
```

Preconditions

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

Effects

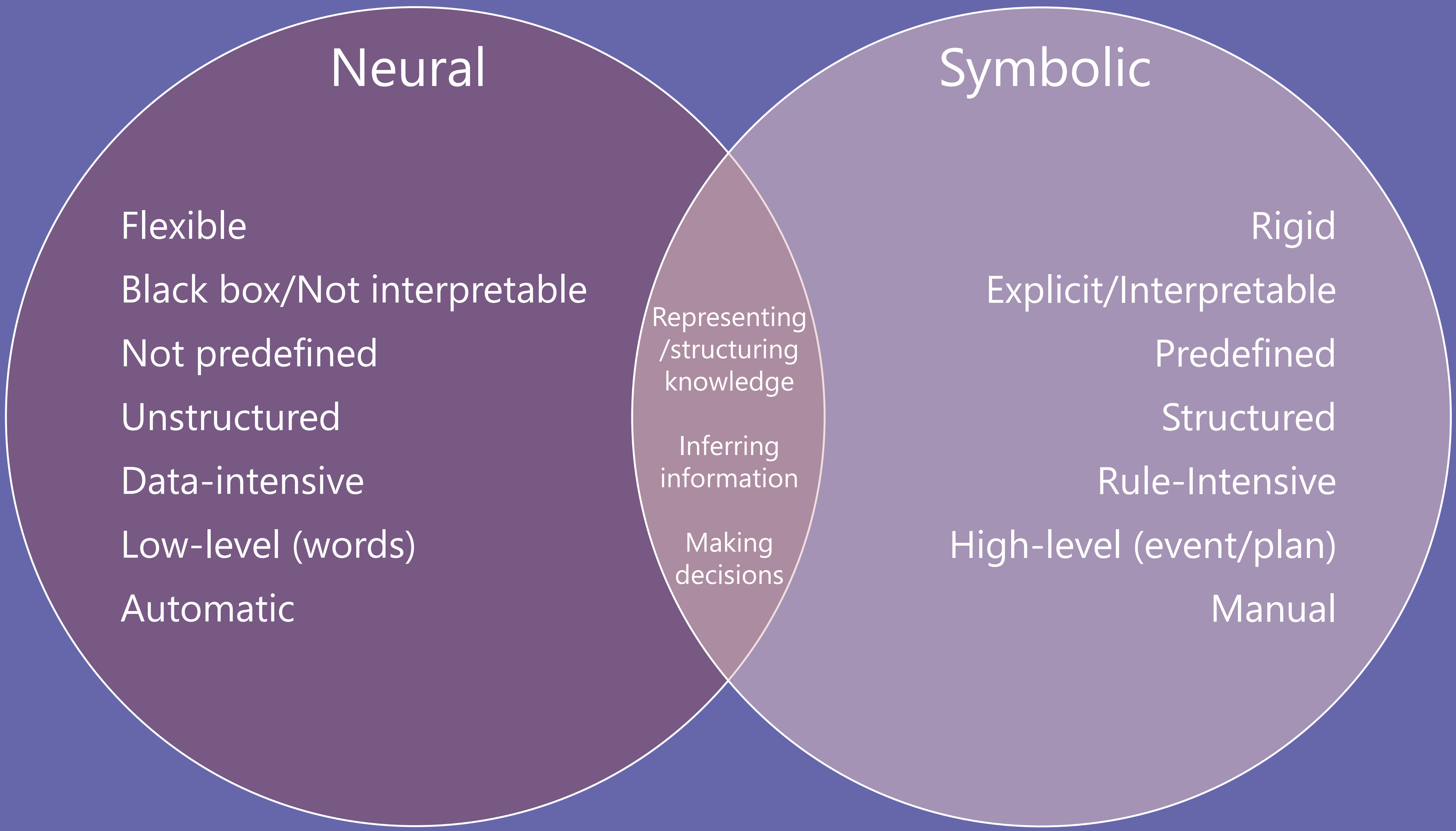
These logical sentences are **literals** – positive or negated atomic sentences

HW 4

Homework 4: Creating Sabre Problems

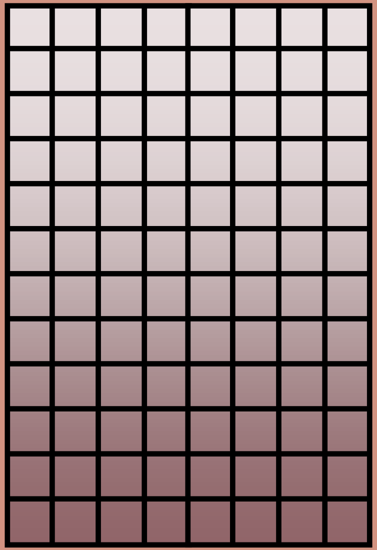
Learning Objectives

- 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

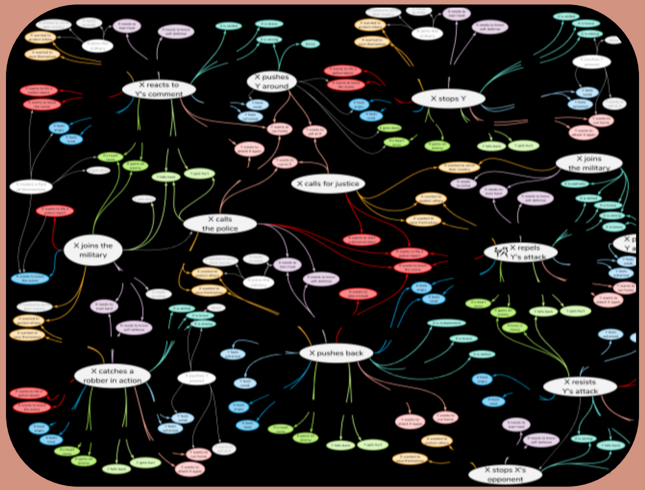

Neural
Symbolic
Flexible
Black box/Not interpretable
Not predefined
Unstructured
Data-intensive
Low-level (words)
Automatic
Representing /structuring knowledge
Inferring information
Making decisions
Rigid
Explicit/Interpretable
Predefined
Structured
Rule-Intensive
High-level (event/plan)
Manual

Solution Outline

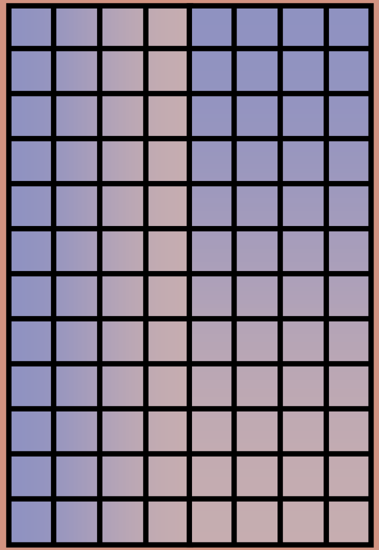
- Leverage manually curated commonsense knowledge resources
- Learn from the examples to induce new relationships
- Scale up using language resources



Learn word embeddings from language corpus



Retrofit word embeddings on semantic resource



Learn knowledge-aware embeddings

Faruqui et al., 2015, Speer et al., 2017

Transfer Learning from Language