Markov Decision Processes

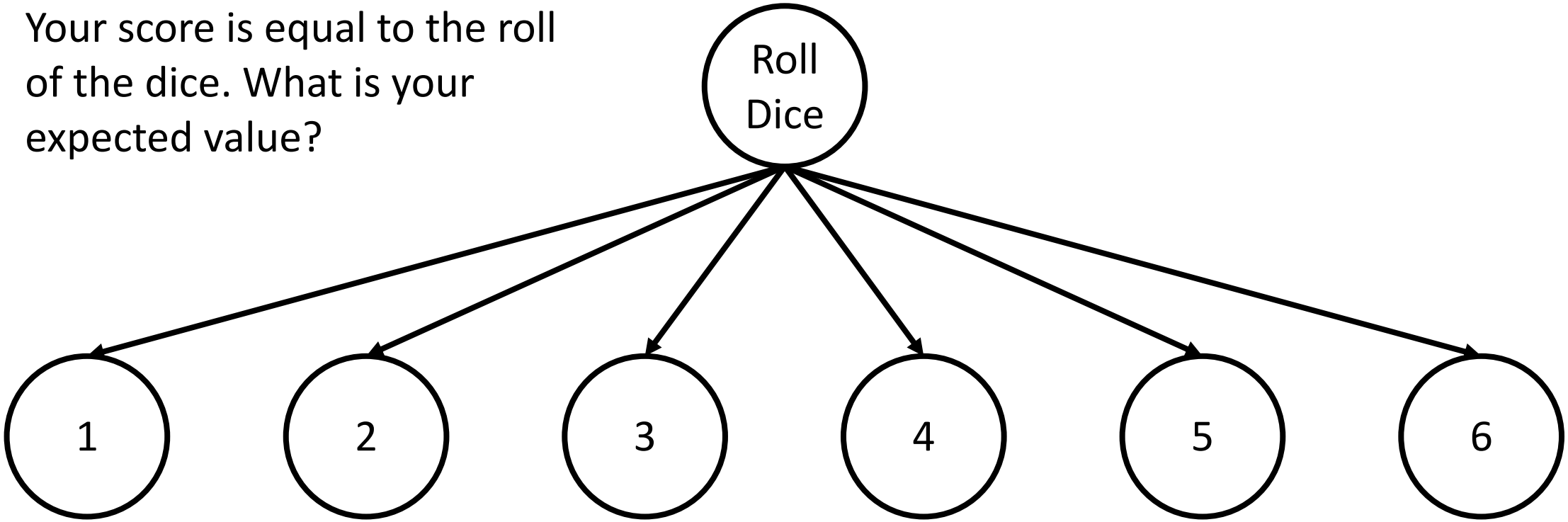
Week 6

Recap

- Trees
  - Deterministic
  - Fully Observable
  - Search
- Min Max Trees
  - Deterministic
  - Fully Observable
  - Expected Value

What about Probabilistic Reasoning?

Your score is equal to the roll of the dice. What is your expected value?



Expected Value

$$E(x) = \sum_{i=0}^{k} p_i x_i$$

$$E(dice) = \frac{1}{6}(1) + \frac{1}{6}(2) + \frac{1}{6}(3) + \frac{1}{6}(4) + \frac{1}{6}(5) + \frac{1}{6}(6)$$

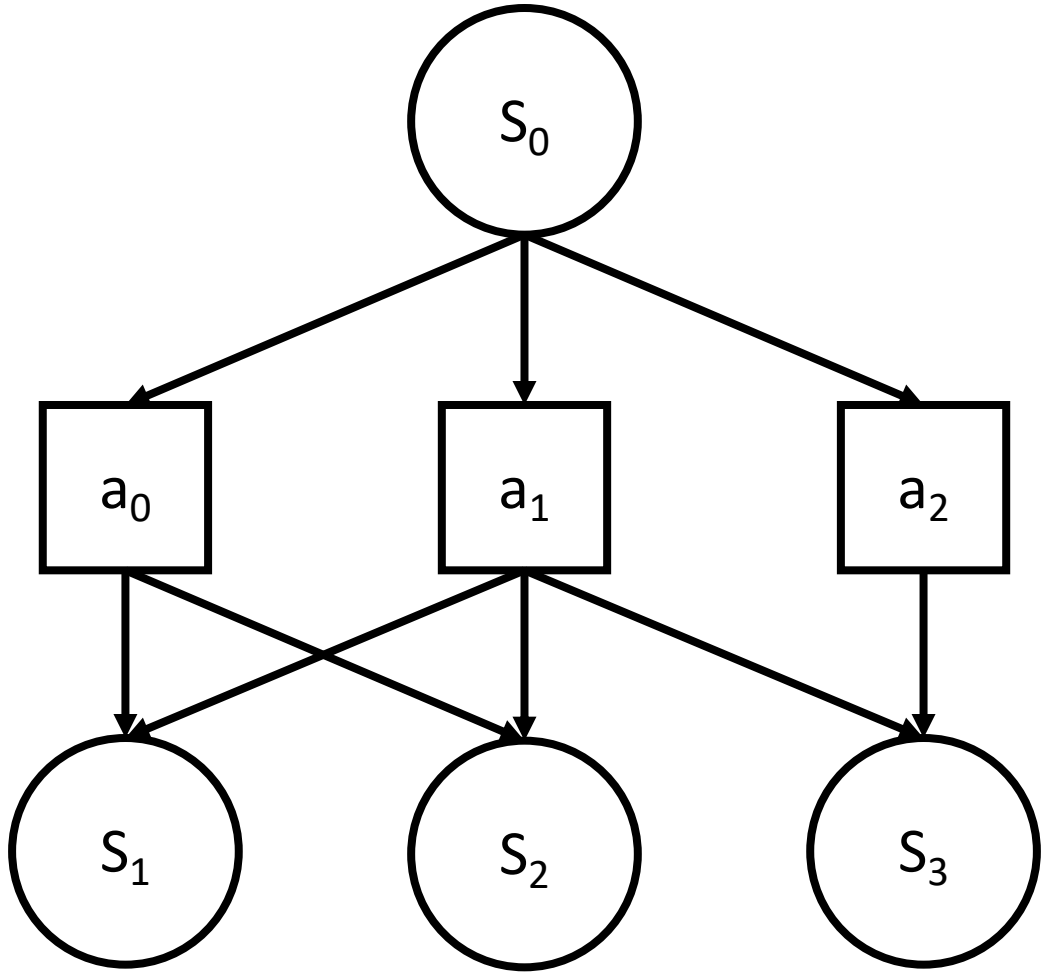
$$E(dice) = \frac{1}{6} + \frac{2}{6} + \frac{3}{6} + \frac{4}{6} + \frac{5}{6} + \frac{6}{6} = \frac{21}{6} = 3.5$$

Probabilistic Actions

- You double your money if you roll a 5 or higher.
  - Actions – Roll, Don't Roll

$$E(x) = \frac{2}{6}(2x) = \frac{2x}{3}$$

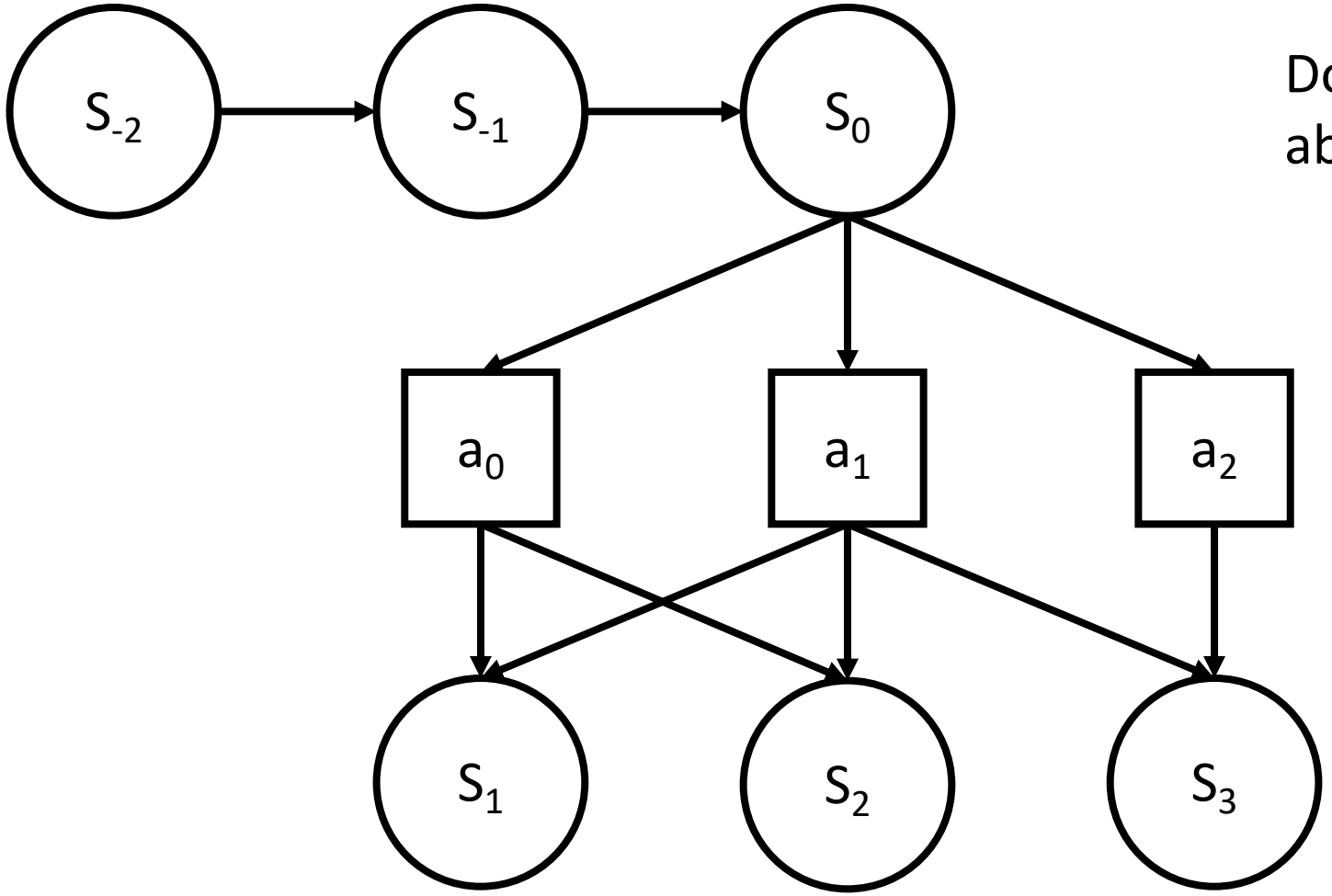
Markov Decision Processes



$$S_0 = 10$$
$$S_1 = 3$$
$$S_2 = 20$$
$$S_3 = 12$$
$$P(S_1|a_0) = 0.4$$
$$P(S_2|a_0) = 0.6$$
$$P(S_1|a_1) = 0.4$$
$$P(S_2|a_1) = 0.3$$
$$P(S_3|a_1) = 0.3$$
$$P(S_3|a_2) = 1$$

Find the rank order of the actions.

Markov Decision Processes

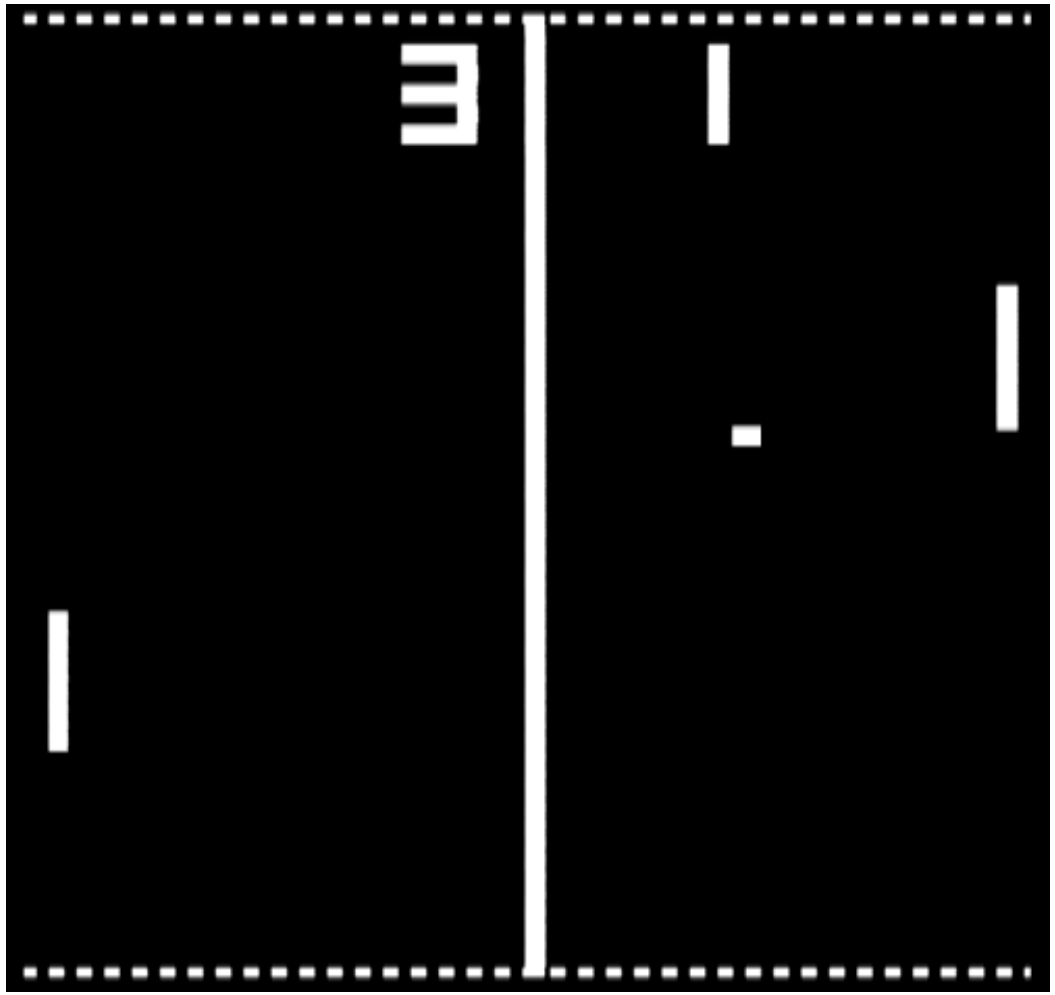


Do we need information about past states?

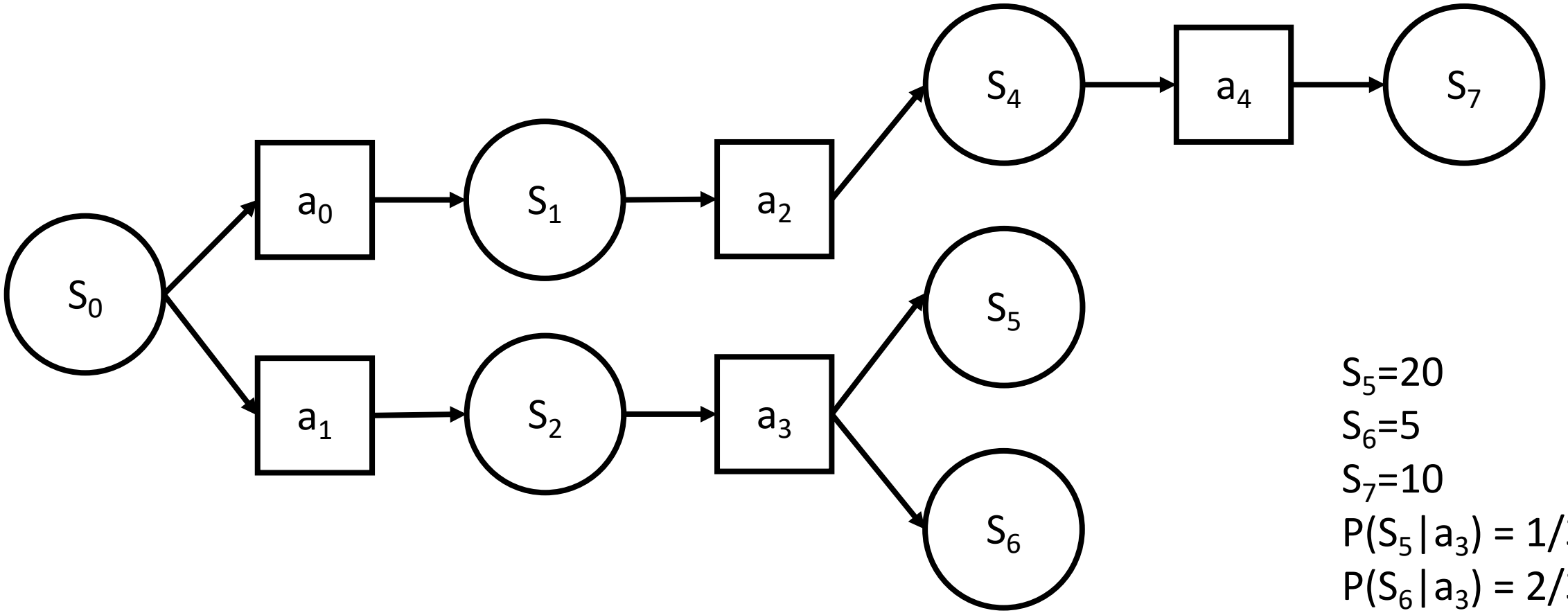
Markov Assumption

- “The future is independent of the past given the present."
  - The only necessary prior information is the current state.
- Often Valid
  - Chess
  - Jeopardy
  - Stock Market
- Violations
  - Video Games
  - Natural Language Generation

Salvaging Markov Assumptions



Rewards in MDPs



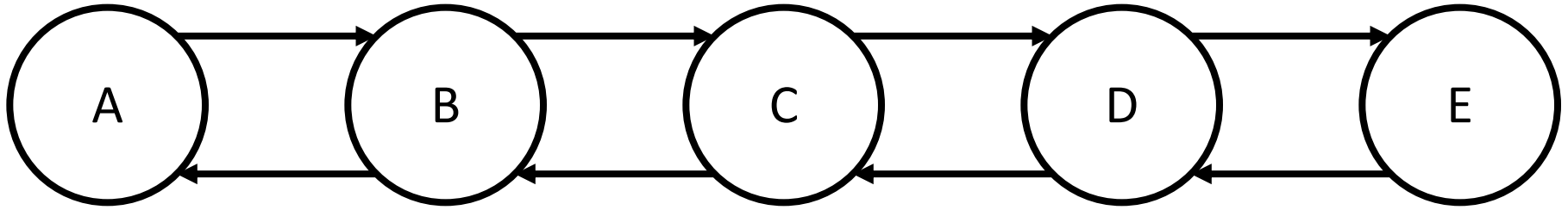
S_5=20
S_6=5
S_7=10
$P(S_5|a_3) = 1/3$
$P(S_6|a_3) = 2/3$

Rewards in MDPs

- What should we prefer?
  - [1,3,5] or [1,2,3]
  - [1,0,0] or [0,0,1]
- Discount future rewards the further they are.
  - Discount factor γ

$$E(s_i) = r_0 + \gamma r_1 + \gamma^2 r_2 \dots \gamma^n r_n = \sum_{k=0}^{n} \gamma^k r_k$$

Rewards in MDPs



R = 10

R = 1

- What is the optimal policy when γ=1?
- What is the optimal policy when γ=0.1?
- What value of γ would give state D equivalent reward from both sides?

Bellman Equation

- Optimal Control in Markov Decision Processes

$$V(s_0) = R(s_0) + \gamma P(s_1|a_o)R(s_1)$$

$$V(s_0) = R(s_0) + \gamma P(s_1|a_o)\big(\gamma P(s_2|a_1)R(s_2)\big)$$

$$V(s) = max_a \left(R(s) + \gamma \sum_{s'} P(s'|s,a)V(s') \right)$$

Problems with MDPs

$$V(s) = max_a \left(R(s) + \gamma \sum_{s'} P(s'|s,a) V(s') \right)$$

- Need intermediate rewards for efficient computation
- Discrete action space
- Discrete state space
- Fully observable

Partially Observable Markov Decision Processes

Review

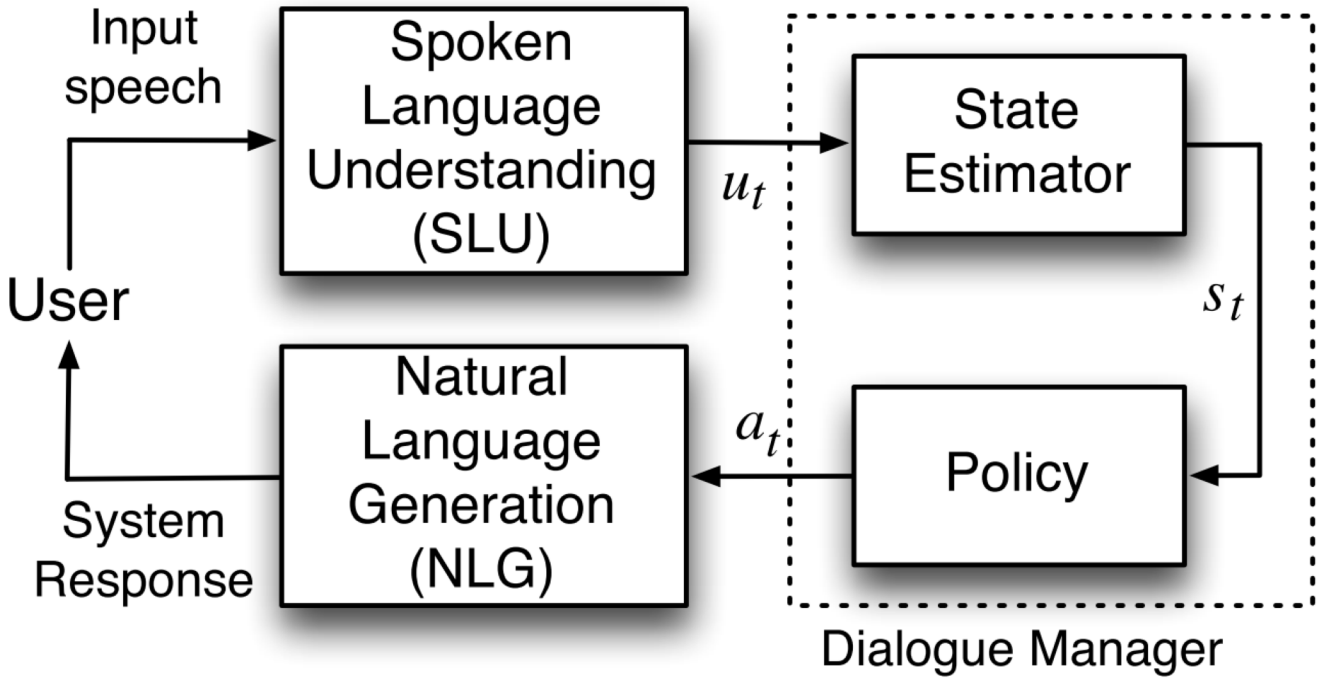


Fig. 1. Components of a finite state-based spoken dialogue system. At each turn the input speech is converted to an abstract representation of the user's intent u_t, the dialogue state s_t is updated and a deterministic decision rule called a *policy* maps the state into an action a_t in response.

POMDP-based Statistical Spoken Dialogue Systems: a Review (Young 2013)

Partially Observable Markov Decision Process

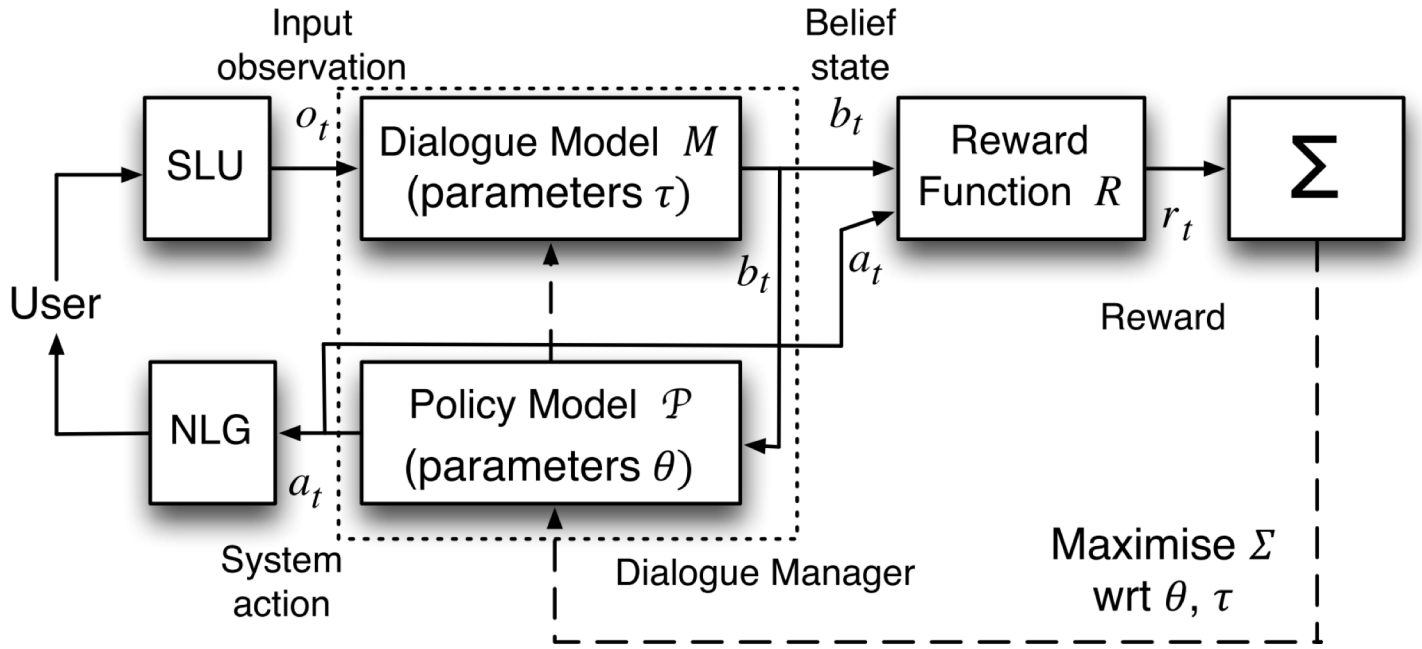
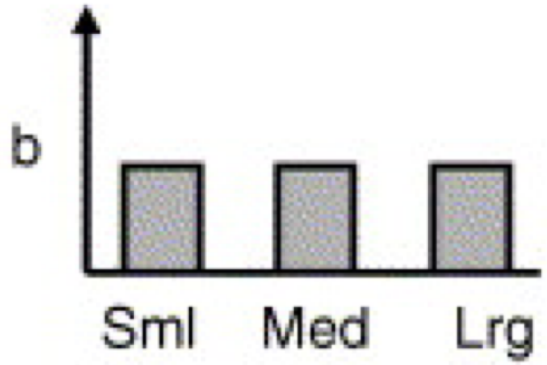
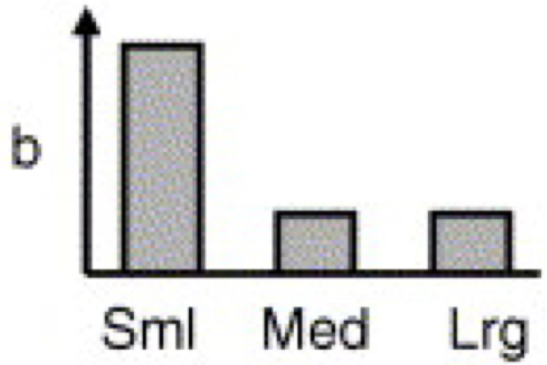


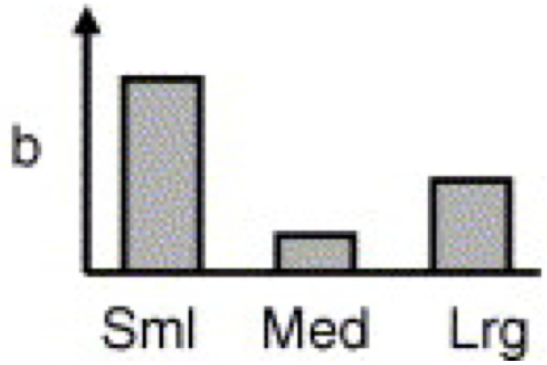
Fig. 2. Components of a POMDP-based spoken dialogue system. In contrast to Fig. 1, the decoded input speech is now regarded as a noisy observation o_t of the underlying user intent u_t. Since u_t is hidden, the system maintains a distribution b_t over all possible dialogue states and instead of trying to estimate the hidden dialogue state, the system response is determined directly from b_t. In addition, the dialogue model and policy are parameterised, and given an appropriate reward function, they can be optimised using reinforcement learning.

Ibid.

System / User / ASR	POMDP belief state	Traditional method

Prior to start of dialog		order: {
		size: <empty>
		…
		}

M: How can I help you?
U: A small pepperoni pizza
[a small pepperoni pizza]
		order: {
		size: small
		…
		}

M: And what type of crust?
U: Uh just normal
[**large** normal]
		order: {
		size: large [?]
		…
		}

POMDP Definition

$$(\mathbb{S}, \mathbb{A}, \mathbb{T}, \mathbb{R}, \mathbb{O}, \mathbb{Z}, \gamma, b_0)$$

S = States

A = Actions

T = Transition Probability

$$P(s_t|s_{t-1}, a_{t-1})$$

R = Expected Rewards

$$r(s_t, a_t) \in \Re$$

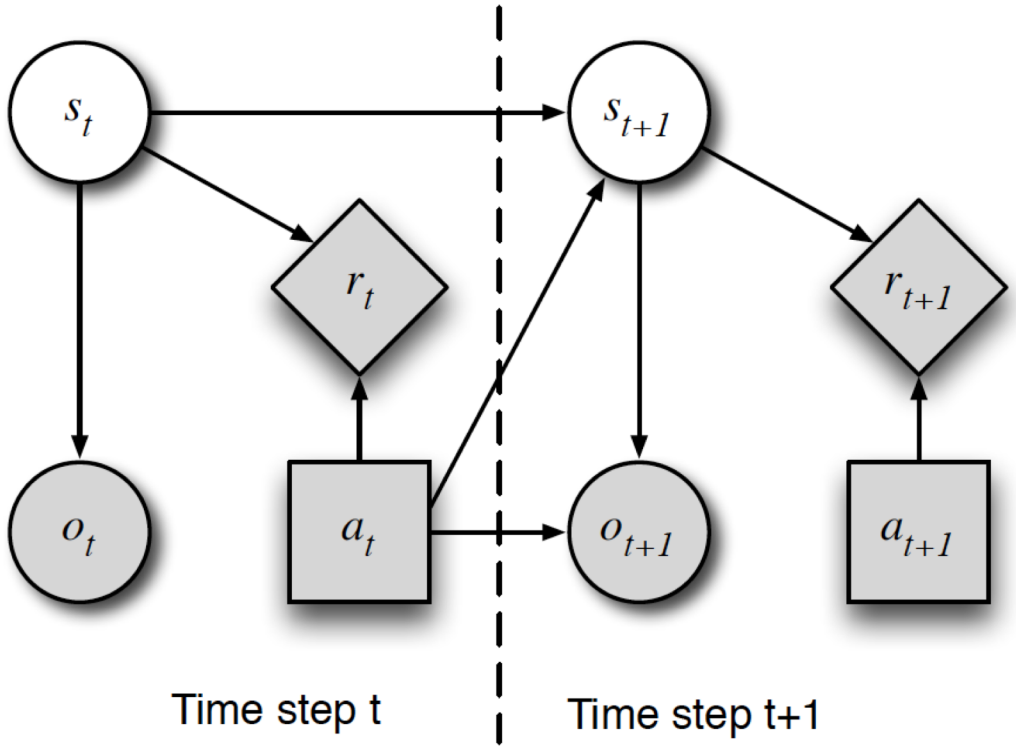
O = Observations

Z = Observation Probability

$$P(o_t|s_t, a_{t-1})$$

γ = Discount Factor

b_0= Initial Belief State



Ibid.

Belief State Representation

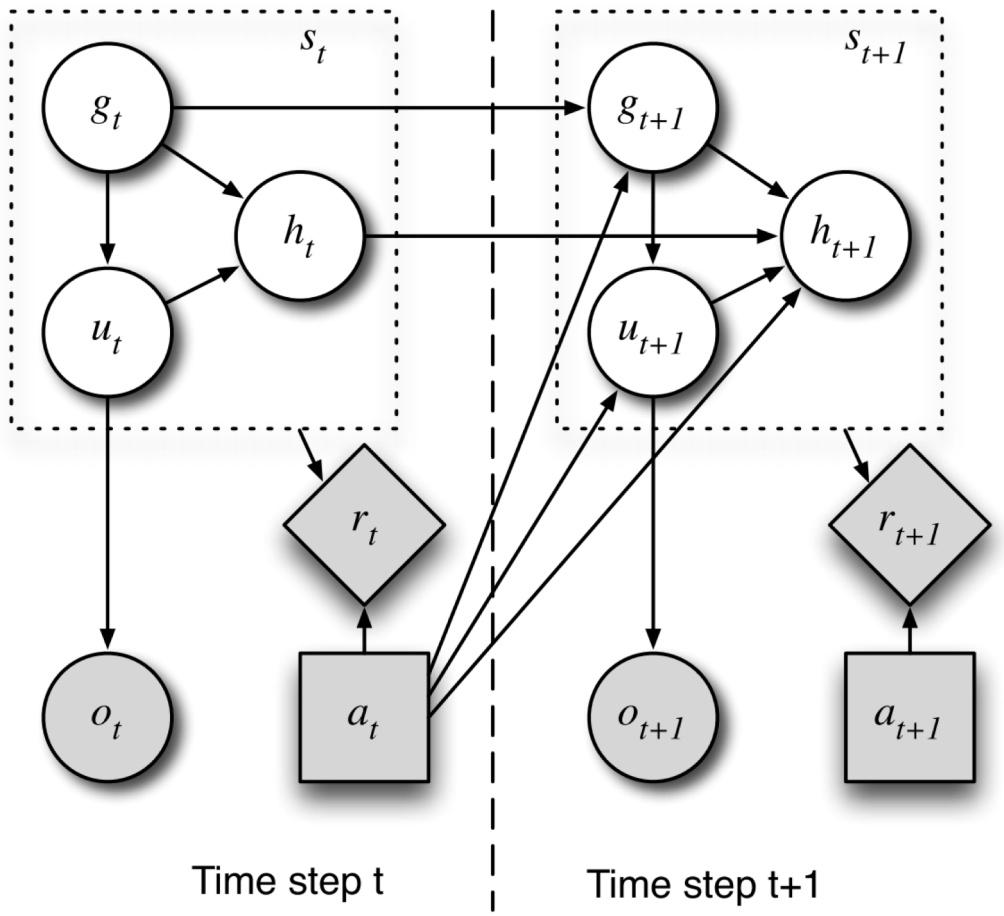
$$s_t = (g_t, u_t, h_t)$$

g = user goal

u = user utterance (intent)

h = history

Still too complex!



Ibid.

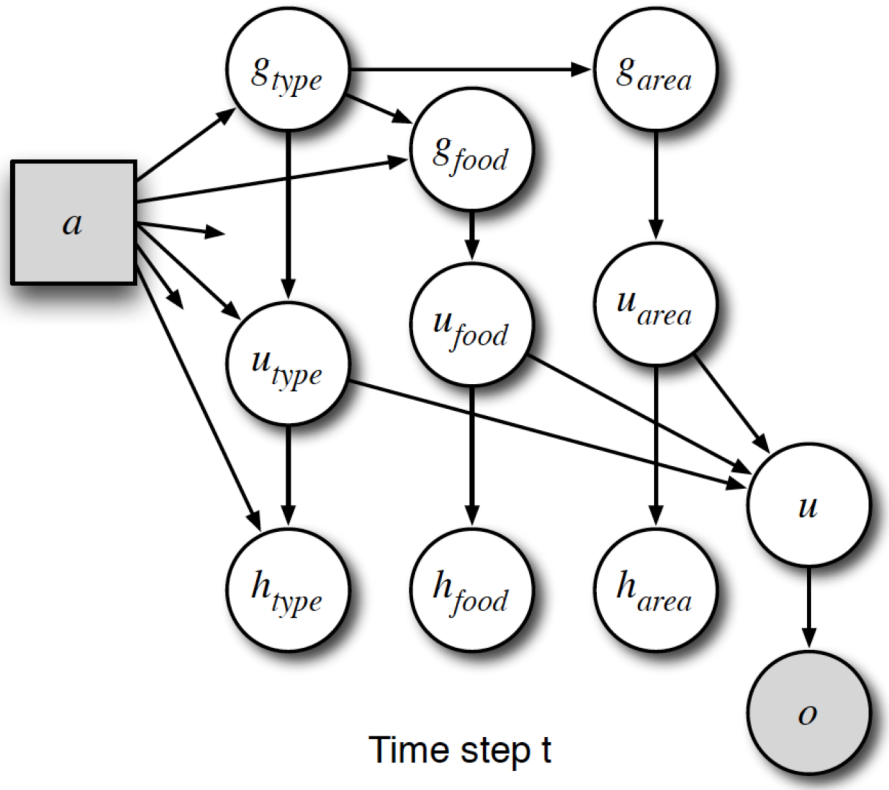
Belief State Representation

$$s_t = (g_t, u_t, h_t)$$

g = user goal

u = user utterance (intent)

h = history



Ibid.

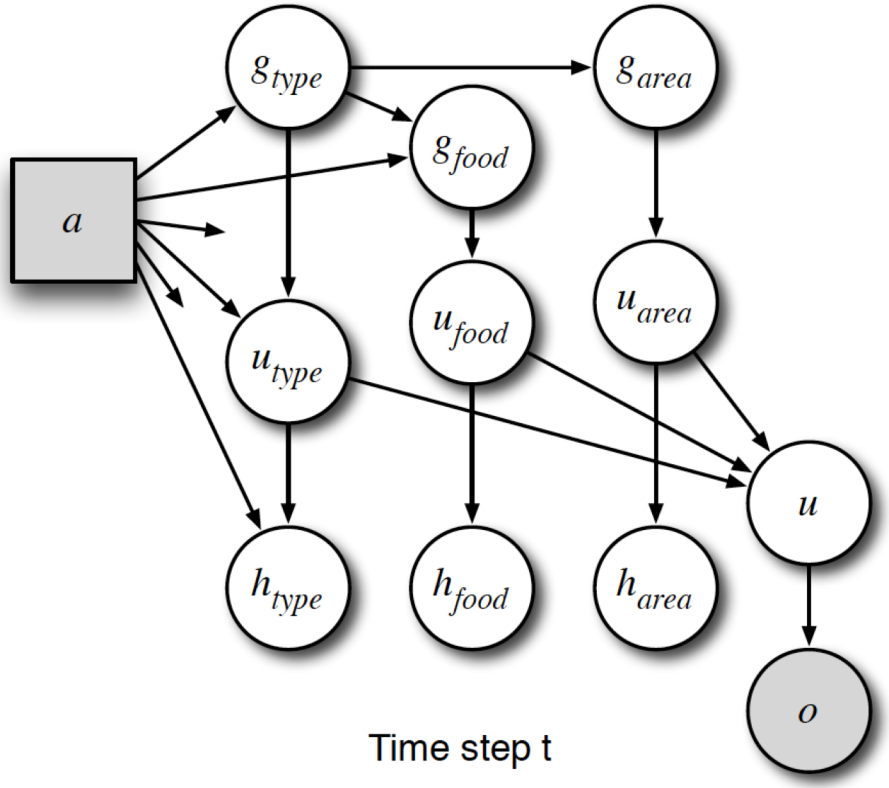
Belief State Representation

$$s_t = (g_t, u_t, h_t)$$

g = user goal

u = user utterance (intent)

h = history

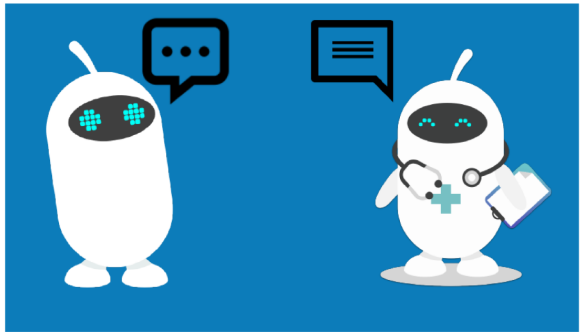


Often still too complex and we must also approximate the policy. Will discuss

Ibid.

User Simulation

It is often difficult to get users to interact with a system particularly during early training iterations.



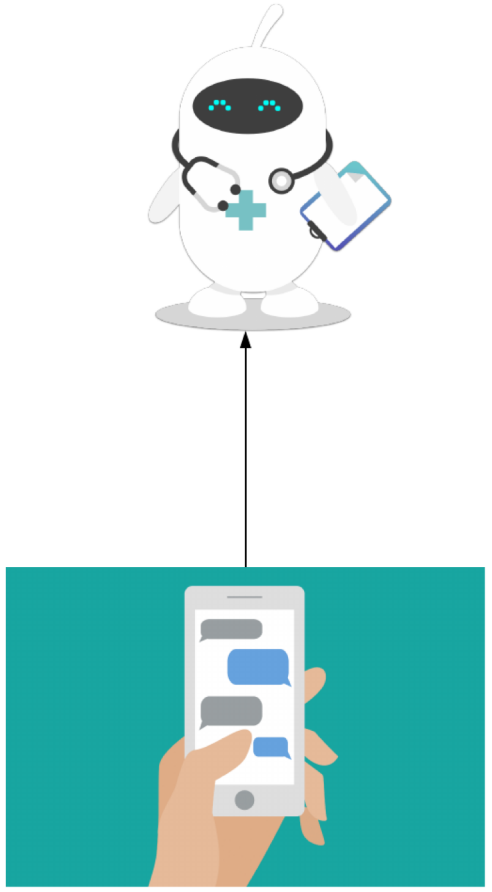
User simulators are often either:

1. Rule-based with a stack-based agenda
2. Stochastic trained on dialog

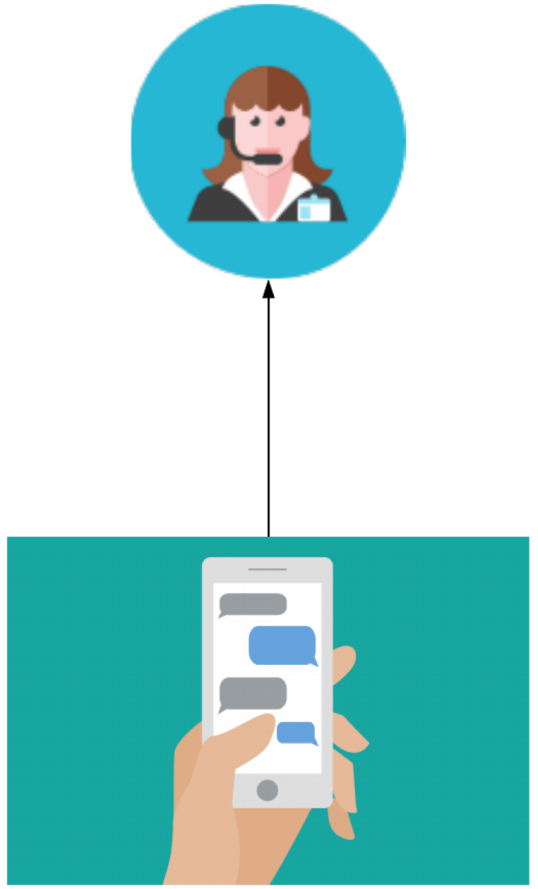
Simulator provides a dialog act at each turn or utterance in the case of some neural language generation models.

Can train indefinitely, but the data is biased. So human data collection is still required, particularly for NLU.

Wizard of Oz



Participant Expectation



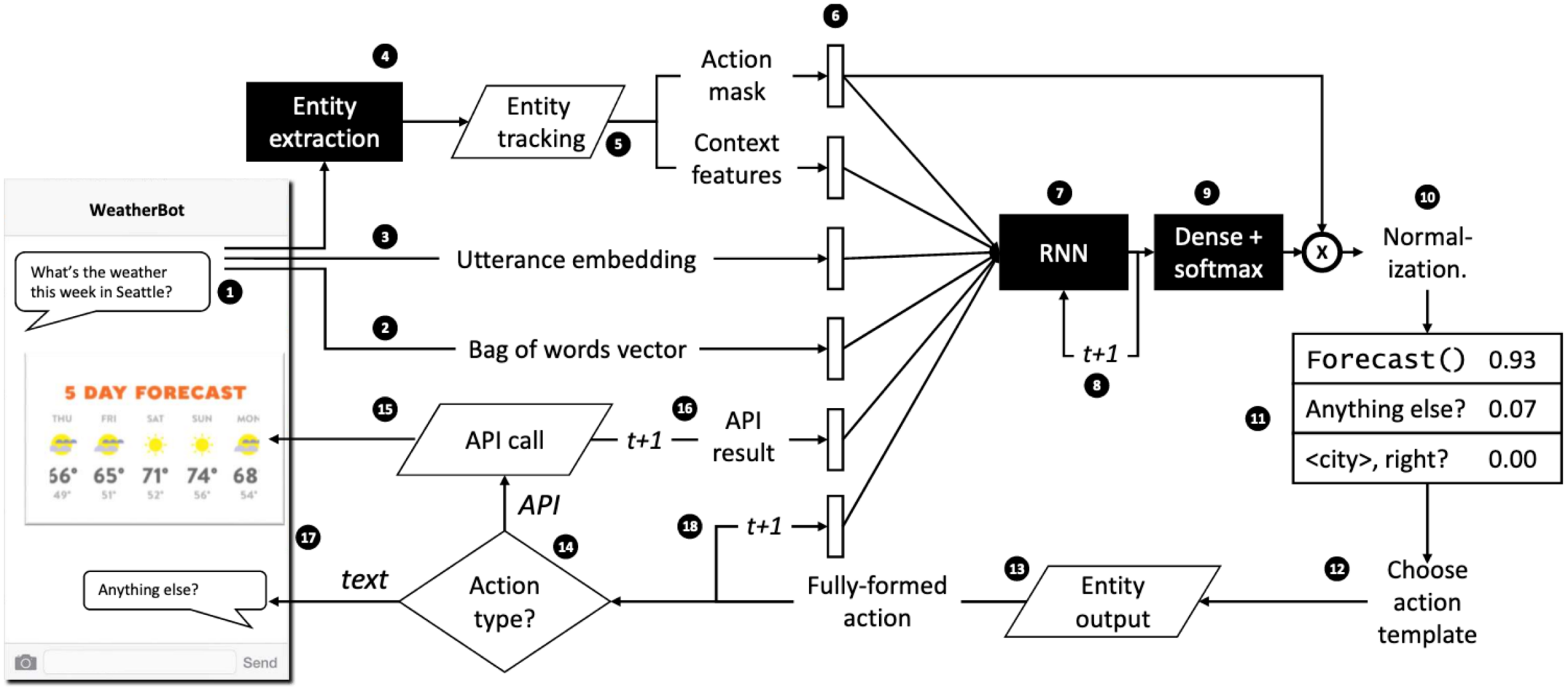
Realit y

Resources

https://github.com/dennybritz/reinforcement-learning

Contains code examples and links to:

- David Silver's Intro to RL on Youtube
- Reinforcement Learning: An Introduction by Sutton and Barto



Hybrid Code Networks (Williams 2017)