

TRACKING



State Tracking
Mar 28th, 2017

Intelligent Conversational Bot

YUN-NUNG (VIVIAN) CHEN WWW.CSIE.NTU.EDU.TW/~YVCHEN/S105-ICB



國立臺灣大學
National Taiwan University

Slides credit from Gašić

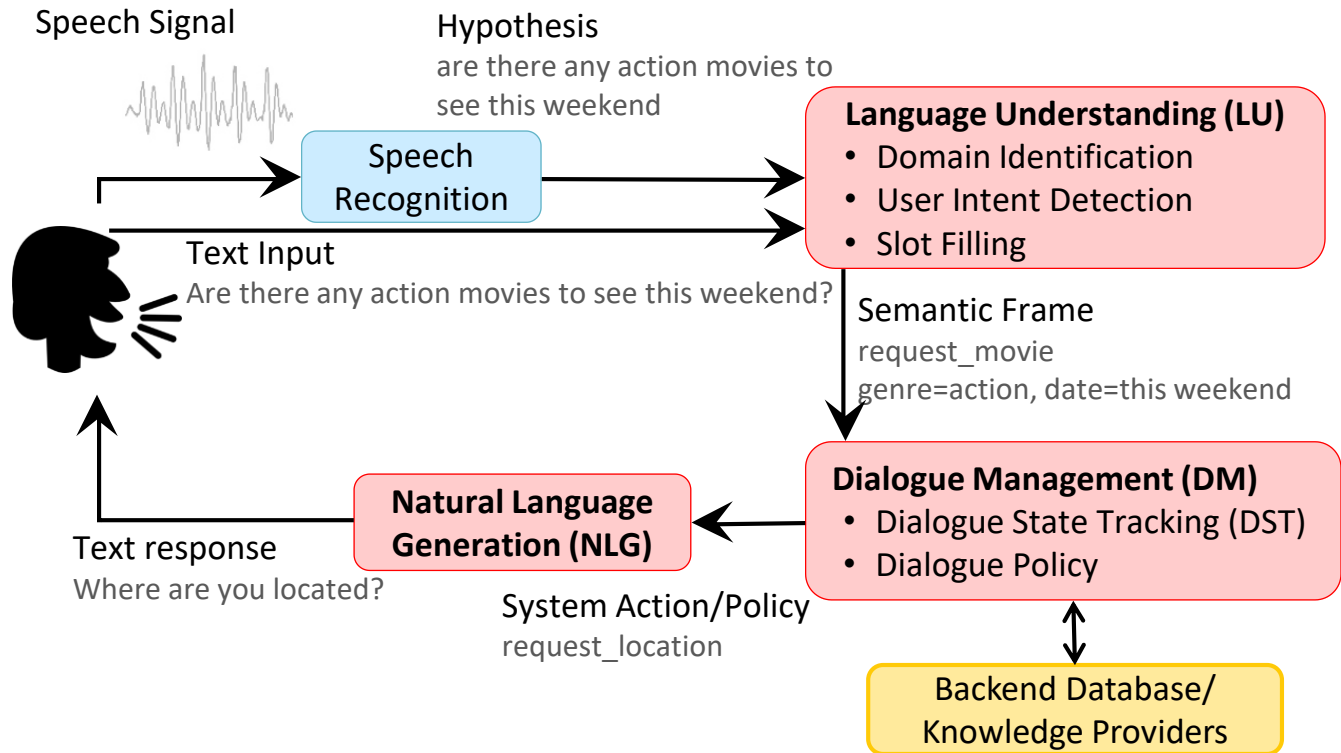
2

Review

Task-Oriented Dialogue System (Young, 2000)

3

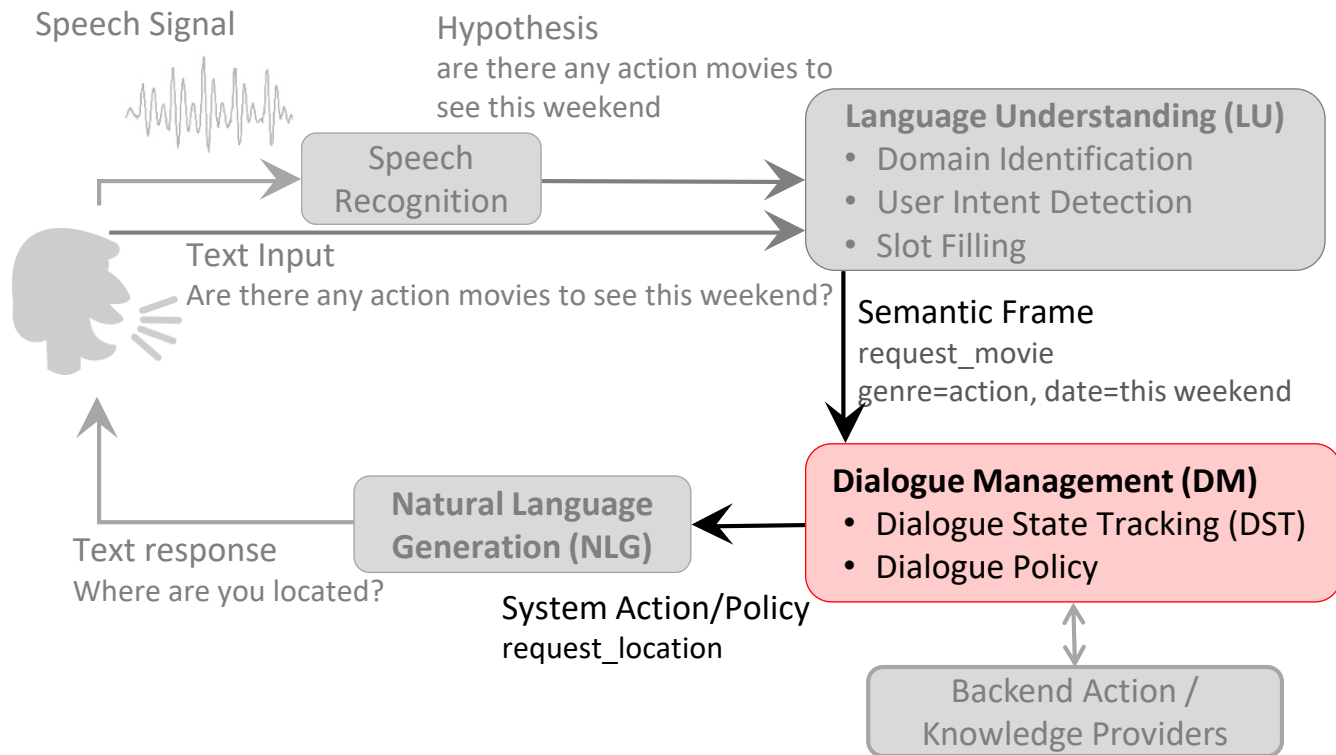
<http://rsta.royalsocietypublishing.org/content/358/1769/1389.short>



Task-Oriented Dialogue System (Young, 2000)

4

<http://rsta.royalsocietypublishing.org/content/358/1769/1389.short>



5

Dialogue Management

Example Dialogue

Hello, how may I help you?

I'm looking for a Thai restaurant.

request (restaurant; foodtype=Thai)

What part of town do you have in mind?

Something in the centre.

inform (area=centre)

Bangkok city is a nice place, it is in the centre of town and it serves Thai food.

What's the address?

request (address)

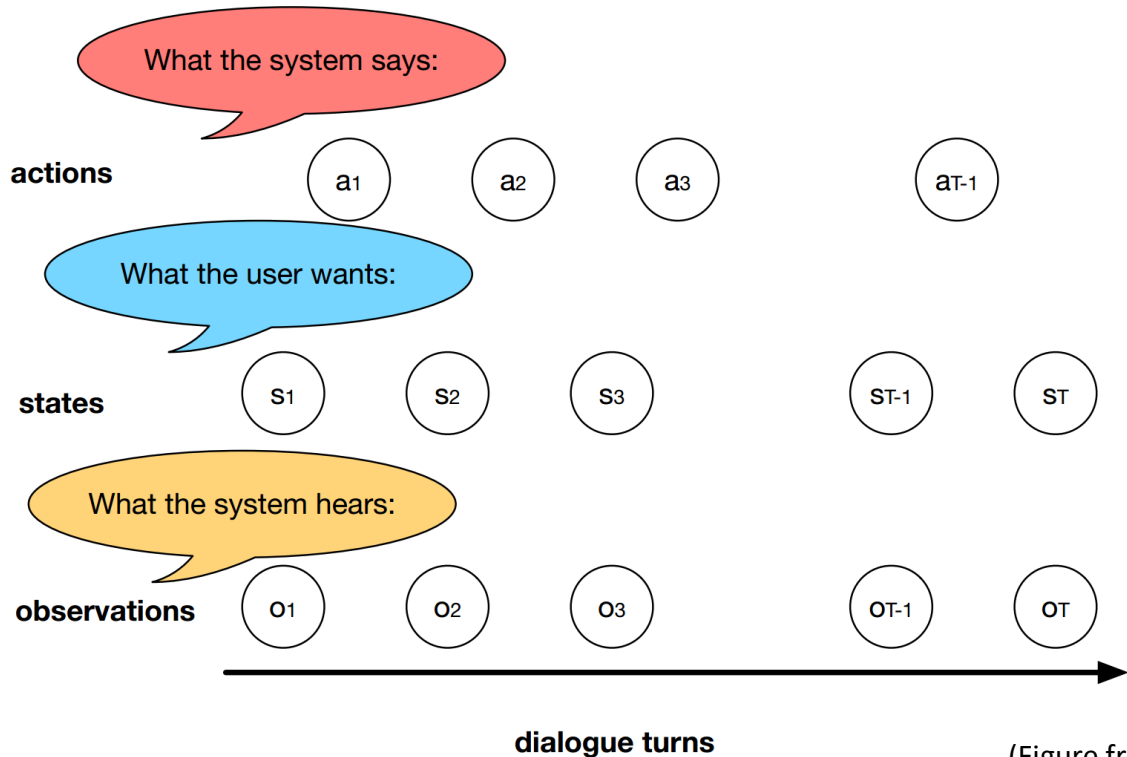
Bangkok city is a nice place, their address is 24 Green street.

Thank you, bye.

bye ()

Elements of Dialogue Management

7



(Figure from Gašić)

Dialogue State Tracking (DST)

- Maintain a probabilistic distribution instead of a 1-best prediction for better robustness to recognition errors

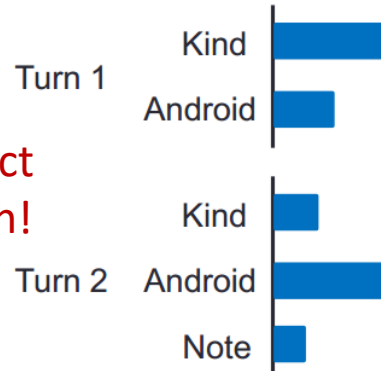
| Turn 1 |
|---------|
| Kind |
| Android |

| Turn 2 |
|---------|
| Note |
| Android |

| Turn 1 | |
|---------|-----|
| Kind | 0.5 |
| Android | 0.3 |

| Turn 2 | |
|---------|-----|
| Note | 0.4 |
| Android | 0.3 |

Incorrect
for both!



Dialogue State Tracking (DST)

- Maintain a probabilistic distribution instead of a 1-best prediction for better robustness to SLU errors or ambiguous input

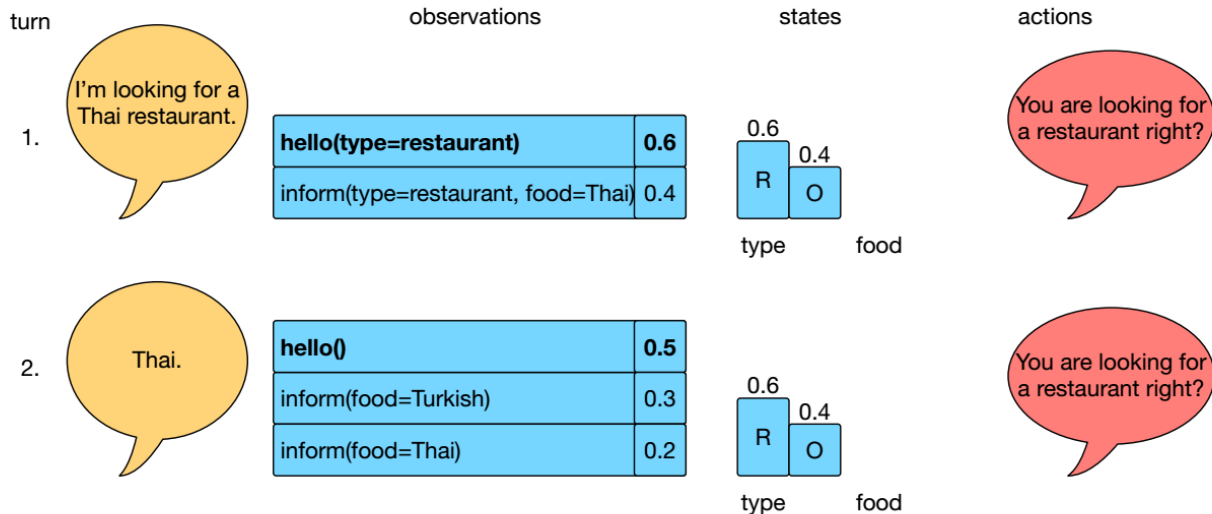
| Slot | Value |
|----------|---------|
| # people | 5 (0.5) |
| time | 5 (0.5) |

| Slot | Value |
|----------|---------|
| # people | 3 (0.8) |
| time | 5 (0.8) |



1-Best Input w/o State Tracking

10



N-Best Inputs w/o State Tracking

11

turn

observations

states

actions

1.

I'm looking for a Thai restaurant.

| | |
|------------------------------------|-----|
| hello(type=restaurant) | 0.6 |
| inform(type=restaurant, food=Thai) | 0.4 |

| | | | |
|-----|-----|-----|-----|
| 0.6 | 0.4 | 0.6 | 0.4 |
| R | O | O | TH |

type

food

What kind of food would you like?

2.

Thai.

| | |
|----------------------|-----|
| hello() | 0.5 |
| inform(food=Turkish) | 0.3 |
| inform(food=Thai) | 0.2 |

| | | | | |
|-----|-----|-----|-----|-----|
| 0.6 | 0.4 | 0.5 | 0.3 | 0.2 |
| R | O | O | TR | TH |

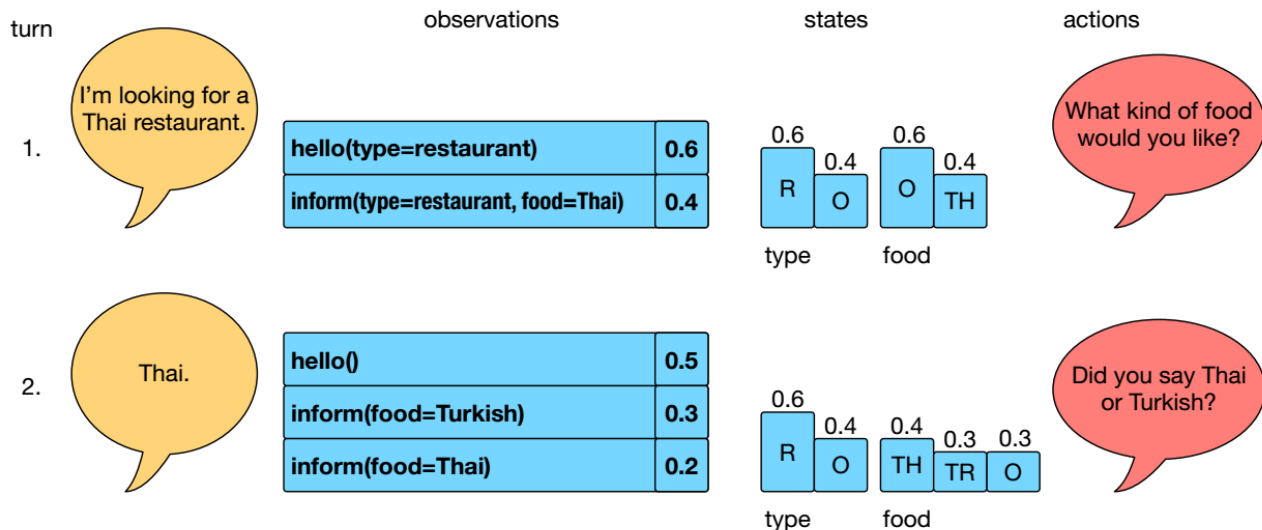
type

food

What kind of food would you like?

N-Best Inputs w/ State Tracking

12



Dialogue State Tracking (DST)

13

- Definition
 - ▣ Representation of the system's belief of the user's goal(s) at any time during the dialogue
- Challenge
 - ▣ How to define the state space?
 - ▣ How to tractably maintain the dialogue state?
 - ▣ Which actions to take for each state?

Define dialogue as a control problem where the behavior can be automatically learned

14

Introduction to RL

Reinforcement Learning

Reinforcement Learning

15

- RL is a general purpose framework for **decision making**
 - ▣ RL is for an *agent* with the capacity to *act*
 - ▣ Each *action* influences the agent's future *state*
 - ▣ Success is measured by a scalar *reward* signal
 - ▣ Goal: *select actions to maximize future reward*

Big three: action, state, reward



Reinforcement Learning

16



Reinforcement Learning

17



Agent learns to take actions to maximize expected reward.

Supervised v.s. Reinforcement

18

□ Supervised

Learning from teacher



"Hello"

Say "Hi"



"Bye bye"

Say "Good bye"

□ Reinforcement



.....



.....

.....

Hello 😊

.....

Learning from critics

Agent

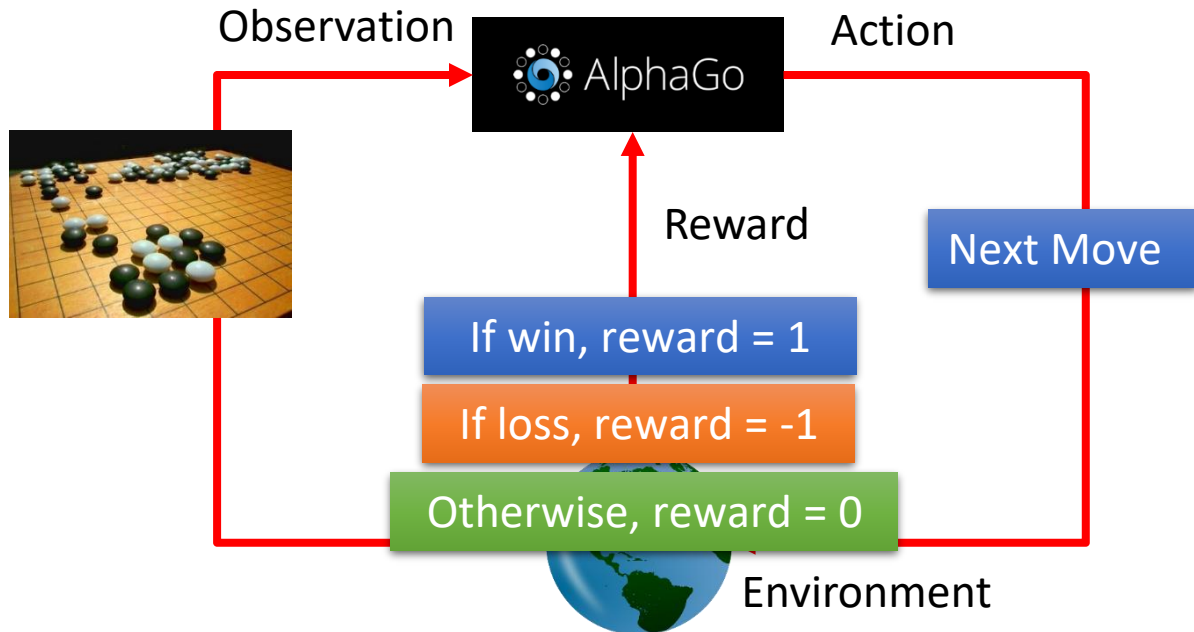
Agent



Bad

Scenario of Reinforcement Learning

19



Agent learns to take actions to maximize expected reward.

RL Based AI Examples

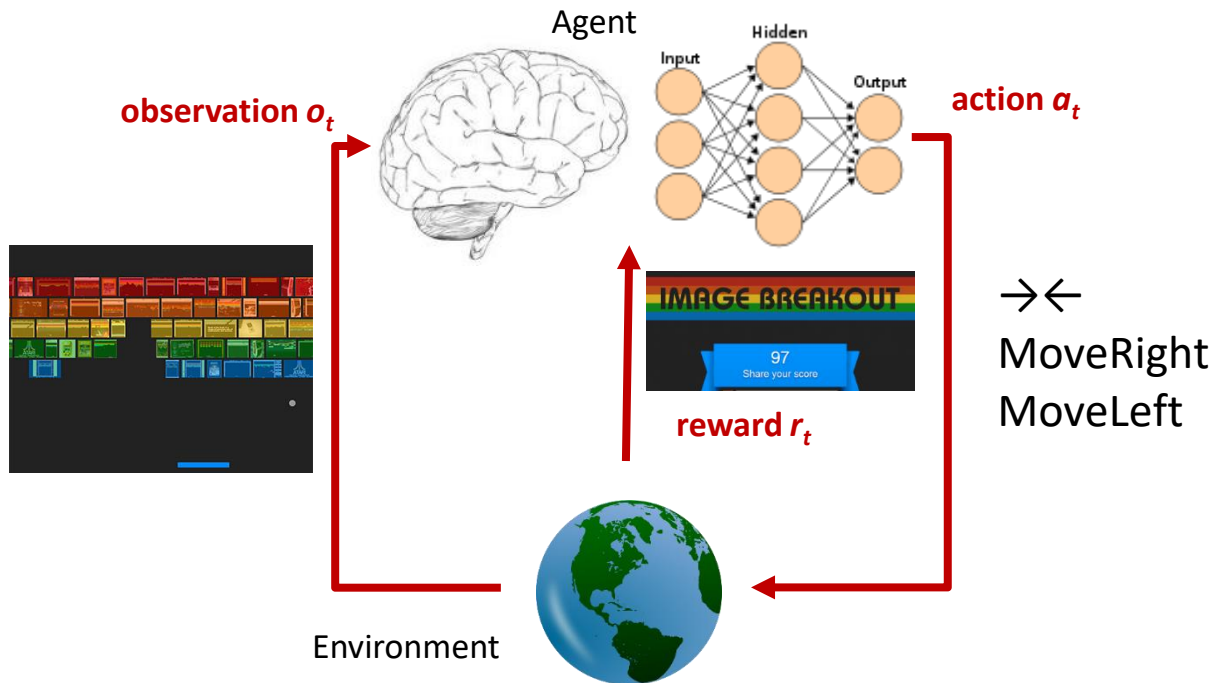
20

- Play games: Atari, poker, Go, ...
- Explore worlds: 3D worlds, ...
- Control physical systems: manipulate, ...
- Interact with users: recommend, optimize, personalize, ...



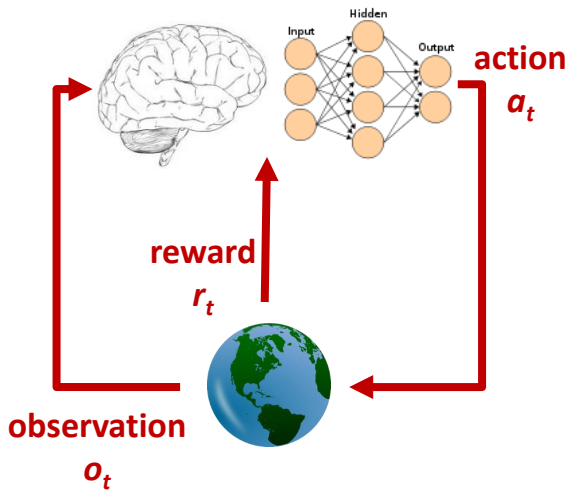
Agent and Environment

21



Agent and Environment

22



□ At time step t

□ The agent

- Executes action a_t
- Receives observation o_t
- Receives scalar reward r_t

□ The environment

- Receives action a_t
- Emits observation o_{t+1}
- Emits scalar reward r_{t+1}

□ t increments at env. step

State

23

- Experience is the sequence of observations, actions, rewards

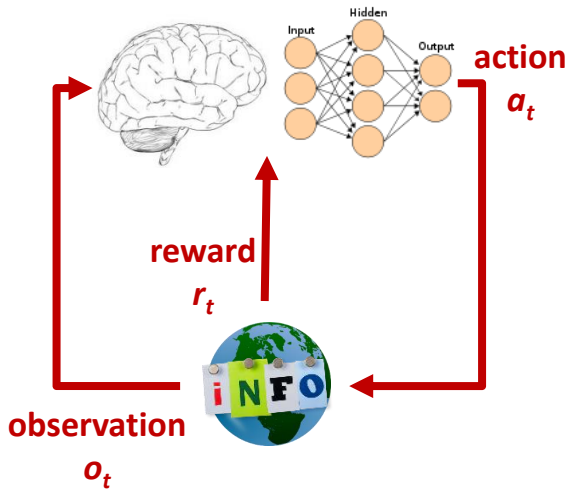
$$O_1, r_1, a_1, \dots, a_{t-1}, O_t, r_t$$

- **State** is the information used to determine what happens next
 - ▣ what happens depends on the history experience
 - The agent selects actions
 - The environment selects observations/rewards
- The state is the function of the history experience

$$s_t = f(o_1, r_1, a_1, \dots, a_{t-1}, o_t, r_t)$$

Environment State

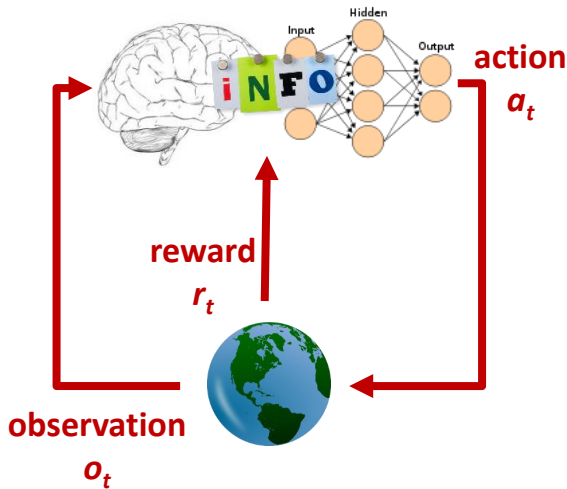
24



- The **environment state** s_t^e is the environment's *private* representation
 - whether data the environment uses to pick the next observation/reward
 - may not be visible to the agent
 - may contain irrelevant information

Agent State

25



- The **agent state** s_t^a is the agent's *internal* representation
 - whether data the agent uses to pick the next action → information used by RL algorithms
 - can be any function of experience

Information State

26

- An information state (a.k.a. Markov state) contains all useful information from history

A state is Markov iff $P(s_{t+1} \mid s_t) = P(s_{t+1} \mid s_1, \dots, s_t)$

- The future is independent of the past given the present

$$H_t = \{o_1, r_1, a_1, \dots, a_{t-1}, o_t, r_t\}$$

$$H_{1:t} \rightarrow s_t \rightarrow H_{t+1:\infty}$$

- Once the state is known, the history may be thrown away
- The state is a sufficient statistics of the future

Fully Observable Environment

27

- Full observability: agent directly observes environment state

$$O_t = s_t^a = s_t^e$$

information state = agent state = environment state

This is a Markov decision process (MDP)

Partially Observable Environment

28

- Partial observability: agent indirectly observes environment

$$s_t^a \neq s_t^e$$

agent state \neq environment state

This is partially observable Markov decision process (POMDP)

- Agent must construct its own state representation s_t^a
 - ▣ Complete history: $s_t^a = H_t$
 - ▣ Beliefs of environment state: $s_t^a = \{P(s_t^e = s^1), \dots, P(s_t^e = s^n)\}$
 - ▣ Hidden state (from RNN): $s_t^a = \sigma(W_s \cdot s_{t-1}^a + W_o \cdot o_t)$

Reward

29

- Reinforcement learning is based on reward hypothesis
- A reward r_t is a scalar feedback signal
 - Indicates how well agent is doing at step t

Reward hypothesis: all agent goals can be desired by maximizing expected cumulative reward

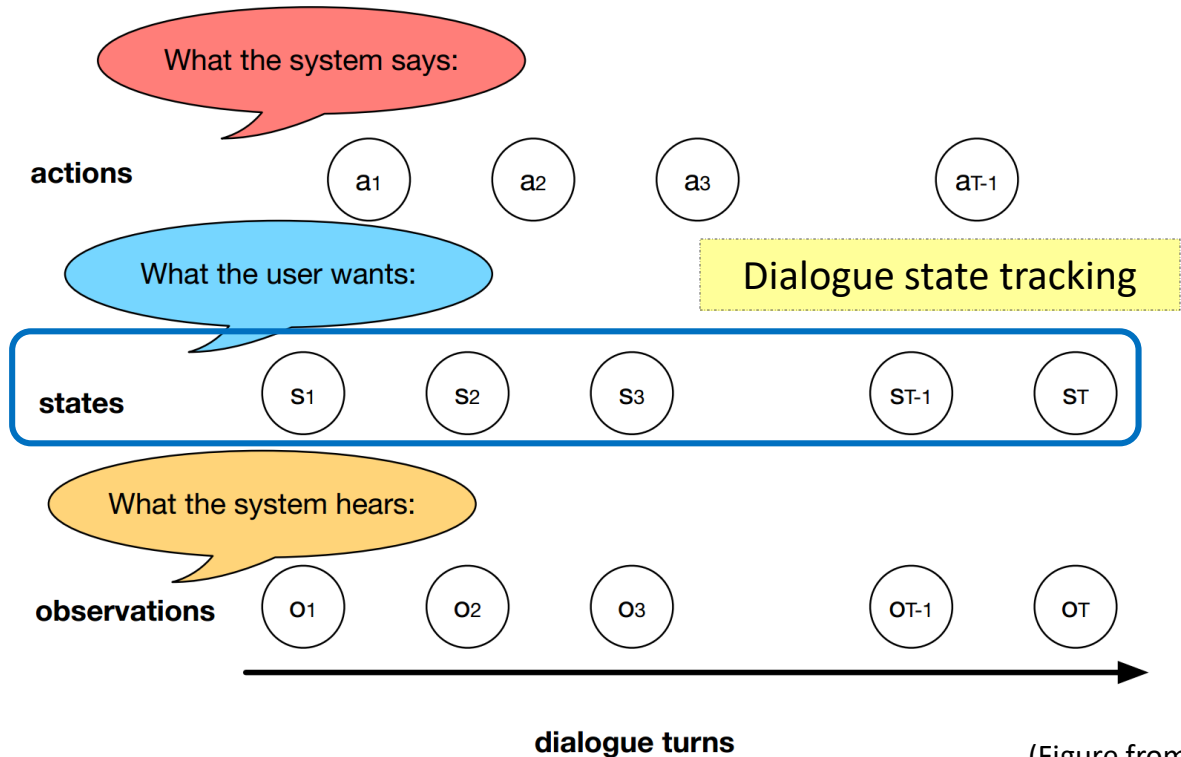
Sequential Decision Making

30

- Goal: select actions to maximize total future reward
 - ▣ Actions may have long-term consequences
 - ▣ Reward may be delayed
 - ▣ It may be better to sacrifice immediate reward to gain more long-term reward



Elements of Dialogue Management



Generative v.s. Discriminative

32

□ Generative

- ▣ The state generates the observation

$$b(s_t) = \frac{p(s_t, o_t)}{\sum_{s_t} p(s_t, o_t)} \propto p(o_t \mid s_t) p(s_t)$$

□ Discriminative

- ▣ The state depends on the observation

$$b(s_t) = p(s_t \mid o_t)$$

33

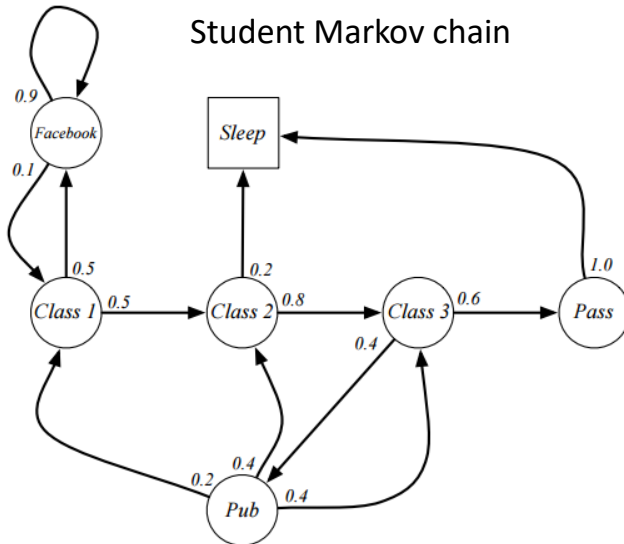
Generative Approach

Dialogue State Tracking

Markov Process

34

- Markov process is a memoryless random process
 - ▣ a sequence of random states S_1, S_2, \dots with the Markov property



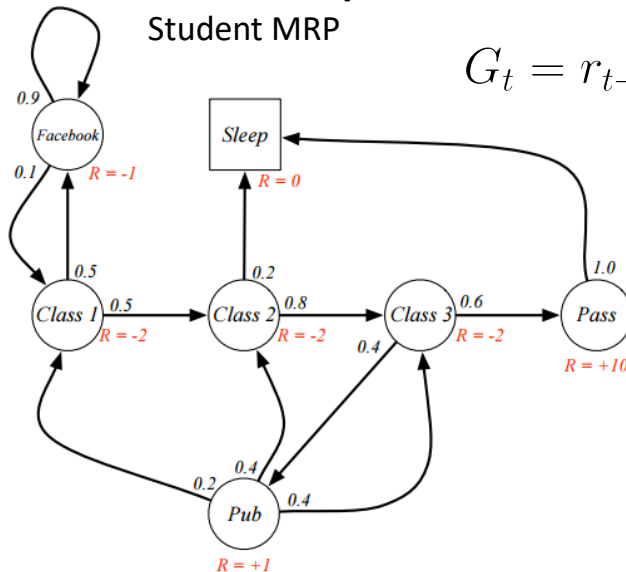
Sample episodes from $S_1 = C1$

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub
- C1 FB FB FB C1 C2 C3 Pub C2 Sleep

Markov Reward Process (MRP)

35

- Markov reward process is a Markov chain with values
 - ▣ The return G_t is the total discounted reward from time-step t

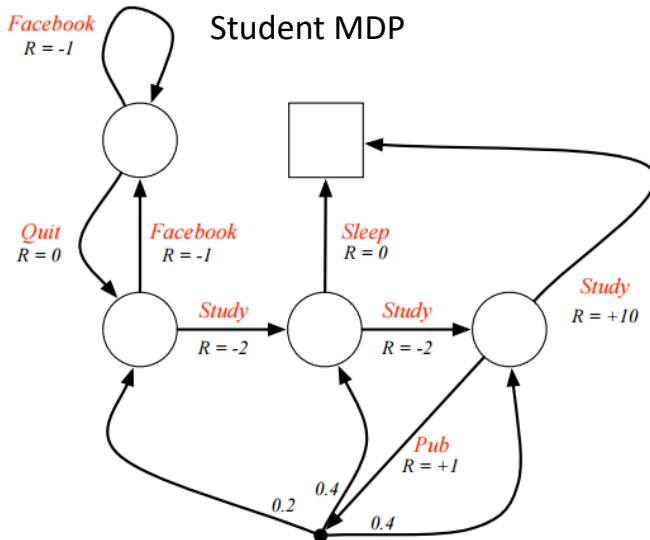


$$G_t = r_{t+1} + \gamma r_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Markov Decision Process (MDP)

36

- Markov decision process is a MRP with decisions
 - ▣ It is an environment in which all states are Markov

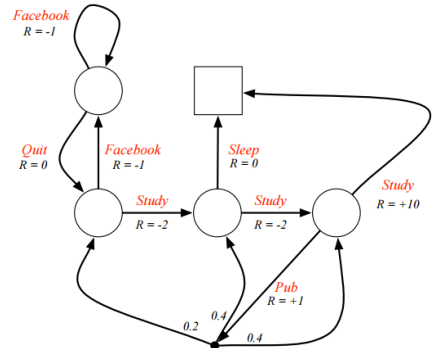


Markov Decision Process (MDP)

37

- S : finite set of **states/observations**
- A : finite set of **actions**
- P : transition **probability**
- R : immediate **reward**
- γ : discount factor
- Goal is to choose **policy** π at time t that maximizes expected overall return:

$$\sum_{t'=t}^T \gamma^{t'-t} r_{t'}$$



DM as Markov Decision Process (MDP)

38



Data

- Dialogue states
- Reward – a measure of dialogue quality



Model

- Markov decision process (MDP)



Prediction

- System actions

DM as Partially Observable Markov Decision Process (POMDP)

39



Data

- Noisy observation of dialogue states
- Reward – a measure of dialogue quality



Model

- Partially observable Markov decision process (POMDP)



Prediction

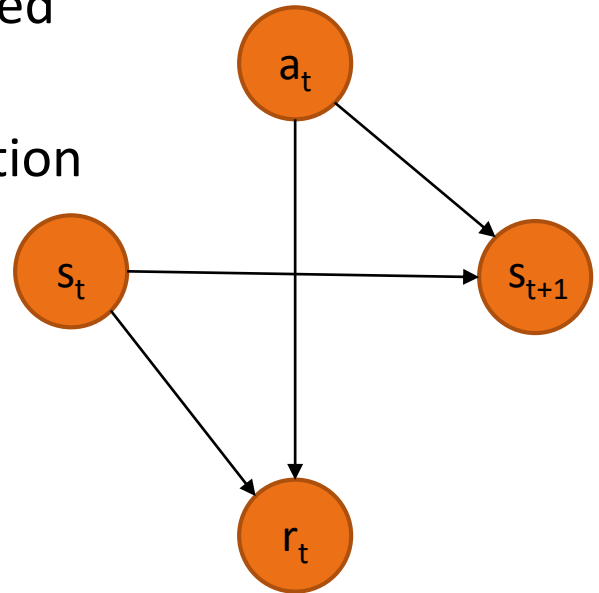
- Distribution over dialogue states
 - Dialogue State Tracking
- Optimal system actions

Markov Decision Process (MDP)

40

- States can be fully observed
- State depends on the previous state and the action

$p(s_{t+1} \mid s_t, a_t)$
transition probability



Partially Observable Markov Decision Process (POMDP)

41

- State generates a noisy observation

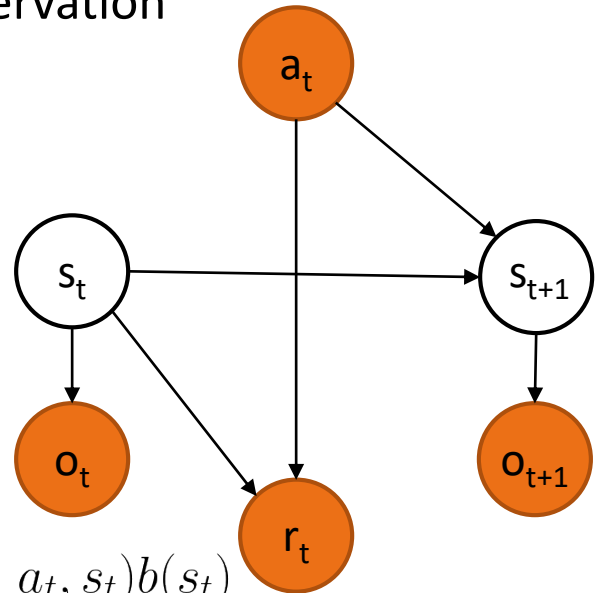
$$p(o_t \mid s_t)$$

observation probability

- State is unobservable and depends on the previous state and the action

$$p(s_{t+1} \mid s_t, a_t)$$

transition probability



$$b(s_{t+1}) \propto p(o_{t+1} \mid s_{t+1}) \sum_{s_t} p(s_{t+1} \mid a_t, s_t) b(s_t)$$

summation over all possible states at every dialogue turn – **intractable!**

Dialogue State Tracking (DST)

42

□ Requirement

□ Dialogue history

- Keep tracking of what happened so far in the dialogue
- Normally done via **Markov property**

□ Task-oriented dialogue

- Need to know what the user wants
- Modeled via the **user goal**

□ Robustness to errors

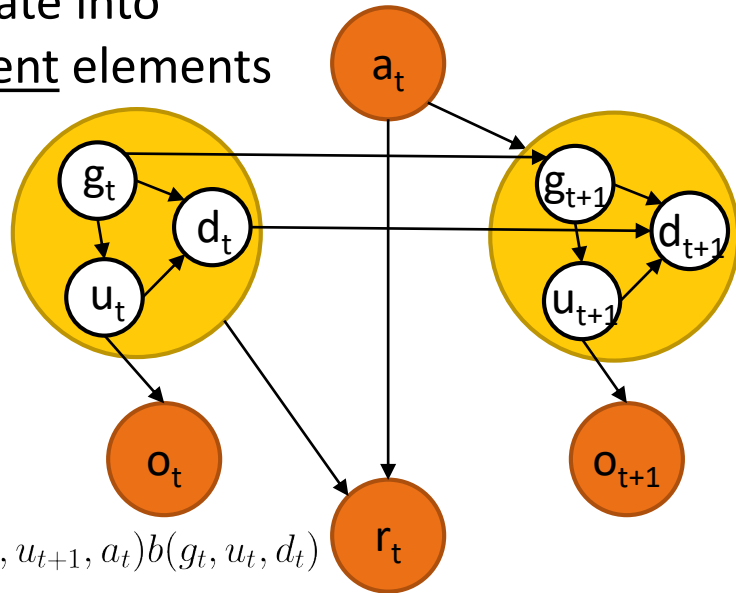
- Need to know what the user says
- Modeled via the **user action**

Dialogue State Factorization

43

- Decompose dialogue state into conditionally independent elements

- User goal g_t
- User action u_t
- Dialogue history d_t



$$b(g_{t+1}, u_{t+1}, d_{t+1}) =$$

$$p(o_{t+1} \mid u_{t+1})p(u_{t+1} \mid g_{t+1}, a_t)$$

$$\sum_{g_t} p(g_{t+1} \mid a_t, g_t) \sum_{d_t, u_t} p(d_{t+1} \mid d_t, g_{t+1}, u_{t+1}, a_t) b(g_t, u_t, d_t)$$

summation over all possible goals – **intractable!**

summation over all possible histories and user actions – **intractable!**

Generative DST

44

- POMDPs are normally intractable for everything
- Two approximations enable POMDP for dialogues
 - I. Hidden Information State (HIS) system (Young et al., 2010)
 - II. Bayesian Update of Dialogue State (BUDS) system (Thomson and Young, 2010)

Hidden Information State (HIS)

45

Observation:
N-best list of
user acts

Ot1

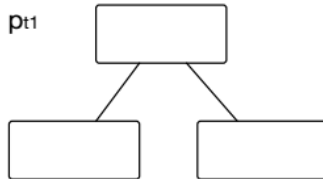
Ot2

•
•
•

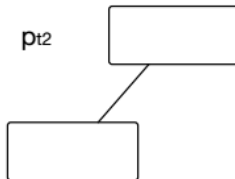
OtN

User Goal:
Partitions of the goal space
built according to ontology

pt1



pt2



**Dialogue
history:**
Grounding
states

dt1

dt2

•
•
•

dtD

Hypotheses:
Every combination of user act,
partition and history

$h1=(Ot1, pt1, dt1)$

$h2=(Ot2, pt1, dt2)$

•
•
•

$h1=(OtN, ptP, dtD)$

Dialogue state: distribution over most likely hypotheses

HIS Partitions

46

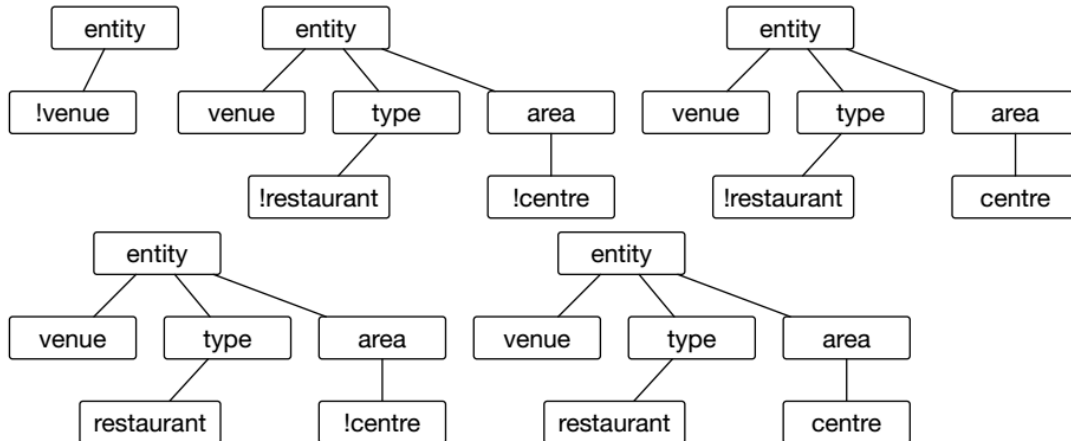
System: How may I help you?
request(task)

User: I'd like a restaurant in the centre.
inform(entity=venue, type=restaurant, area=centre)

entity=venue

area=centre

type=restaurant



Pruning

47

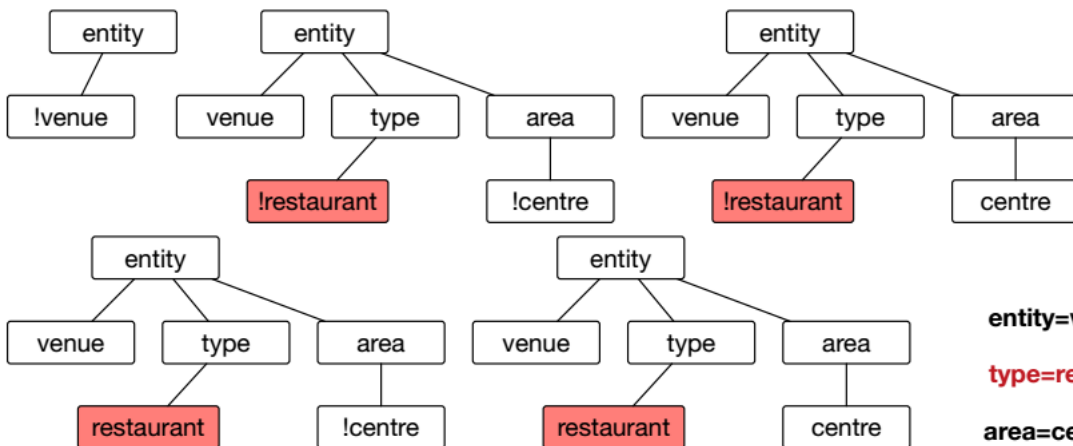
System: How may I help you?
request(task)

User: I'd like a restaurant in the centre.
inform(entity=venue, type=restaurant, area=centre)

entity=venue

area=centre

type=restaurant

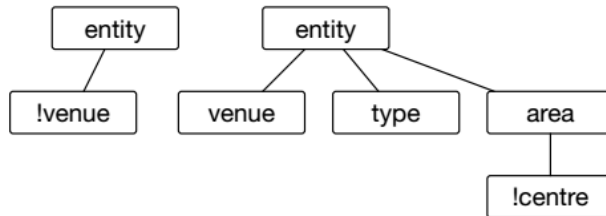


Pruning

48

System: How may I help you?
request(task)

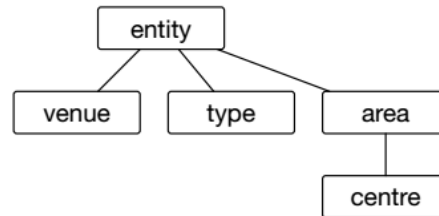
User: I'd like a restaurant in the centre.
inform(entity=venue, type=restaurant, area=centre)



entity=venue

area=centre

type=restaurant



entity=venue 0.9

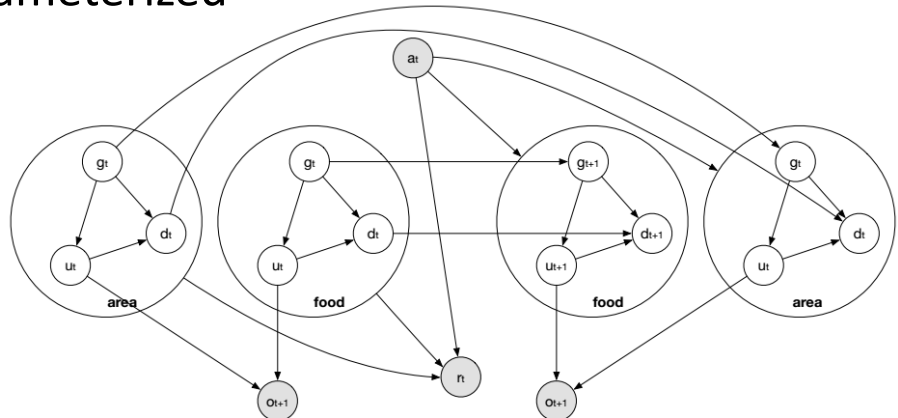
area=centre 0.5

Bayesian Update of Dialogue State (BUDS)

49

□ Idea

- Further decomposes the dialogue state
- Produce tractable state update
- Transition and observation probability distributions can be parameterized



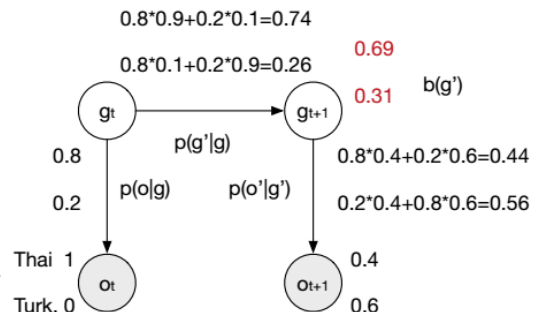
BUDS Belief Tracking

50

- Expectation propagation
 - ▣ Allow parameters tying
 - ▣ Handle factorized hidden variables
 - ▣ Handle large state spaces
- Example

| $p(o g)$ | $o : \text{Thai}$ | $o : \text{Turk.}$ |
|--------------------|-------------------|--------------------|
| $g : \text{Thai}$ | 0.8 | 0.2 |
| $g : \text{Turk.}$ | 0.2 | 0.8 |

| $p(g' g)$ | $g' : \text{Thai}$ | $g' : \text{Turk.}$ |
|--------------------|--------------------|---------------------|
| $g : \text{Thai}$ | 0.9 | 0.1 |
| $g : \text{Turk.}$ | 0.1 | 0.9 |



51

Discriminative Approach

Dialogue State Tracking

Generative v.s. Discriminative

52

□ Generative

- ▣ The state generates the observation

$$b(s_t) = \frac{p(s_t, o_t)}{\sum_{s_t} p(s_t, o_t)} \propto p(o_t | s_t) p(s_t)$$

Assumption: observations at each turn are independent

□ Discriminative

- ▣ The state depends on the observation

$$b(s_t) = p(s_t | o_t)$$

Directly model dialogue states given arbitrary input features

DST Problem Formulation

53

- The DST dataset consists of
 - ▣ *Goal*: for each informable slot
 - e.g. price=cheap
 - ▣ *Requested*: slots by the user
 - e.g. moviename
 - ▣ *Method*: search method for entities
 - e.g. by constraints, by name
- The dialogue state is
 - ▣ the distribution over possible slot-value pairs for goals
 - ▣ the distribution over possible requested slots
 - ▣ the distribution over possible methods

Class-Based DST

54



Data

- Observations labeled w/ dialogue state



Model

- Neural networks
- Ranking models

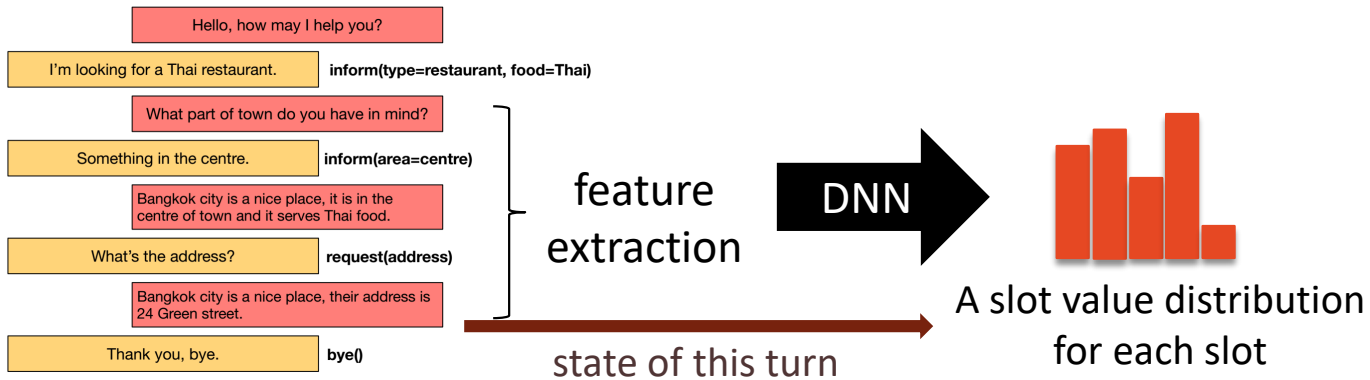


Prediction

- Distribution over dialogue states
 - Dialogue State Tracking

DNN for DST

55



Sequence-Based DST

56



Data

- Sequence of observations labeled w/
dialogue state



Model

- Recurrent neural networks (RNN)



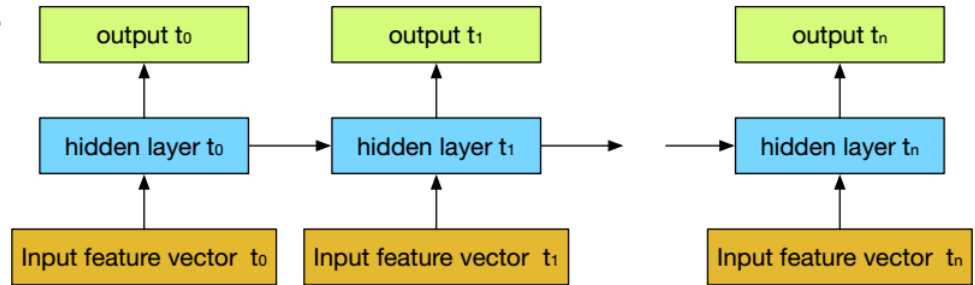
Prediction

- Distribution over dialogue states
 - Dialogue State Tracking

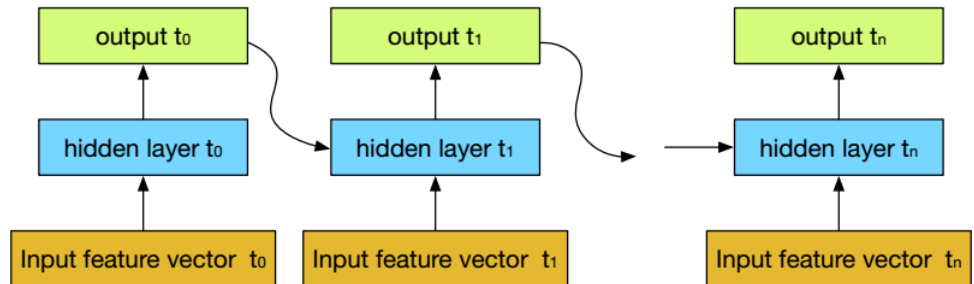
Recurrent Neural Network (RNN)

57

□ Elman-type



□ Jordan-type



RNN DST

58

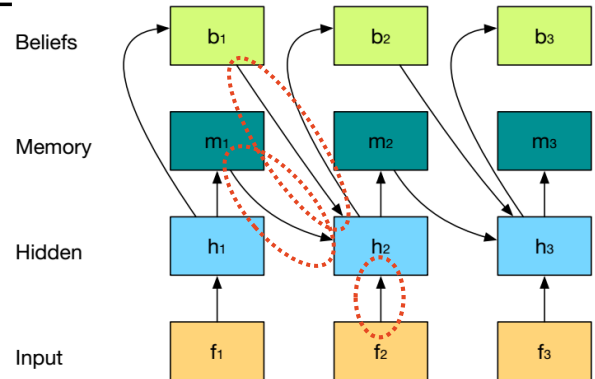
- Idea: internal memory for representing dialogue context

- Input

- most recent dialogue turn
- last machine dialogue act
- dialogue state
- memory layer

- Output

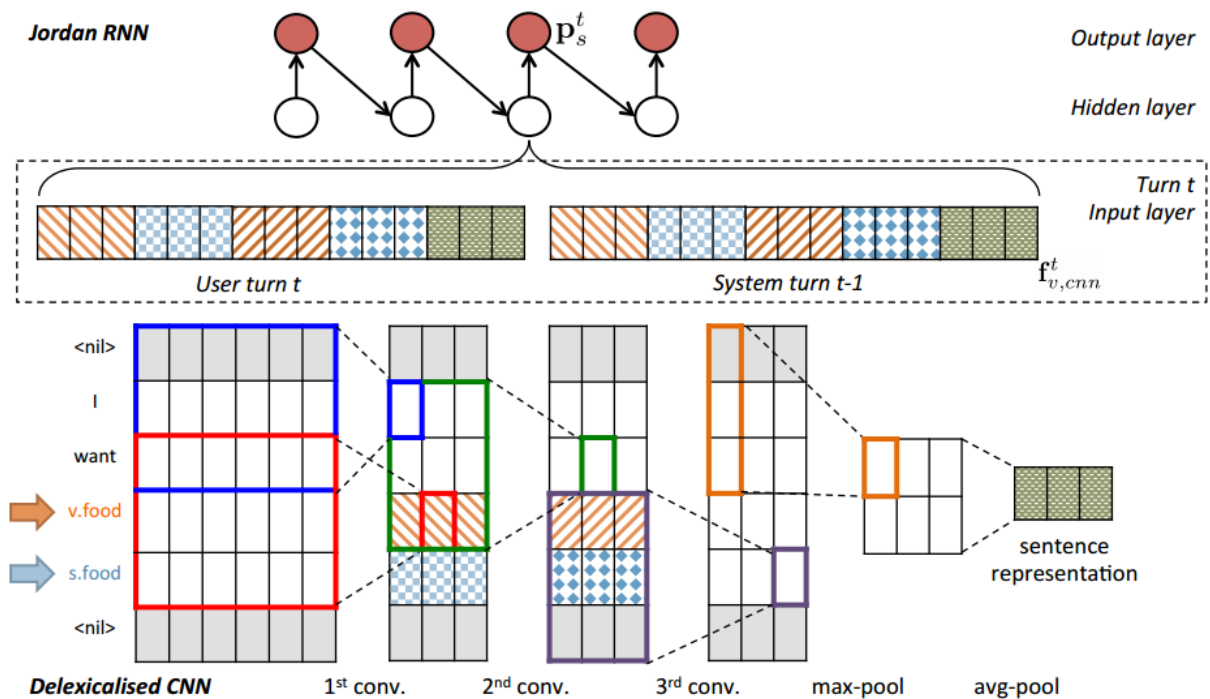
- update its internal memory
- distribution over slot values



RNN-CNN DST

59

<http://www.anthology.aclweb.org/W/W13/W13-4073.pdf>; <https://arxiv.org/abs/1506.07190>



(Figure from Wen et al, 2016)

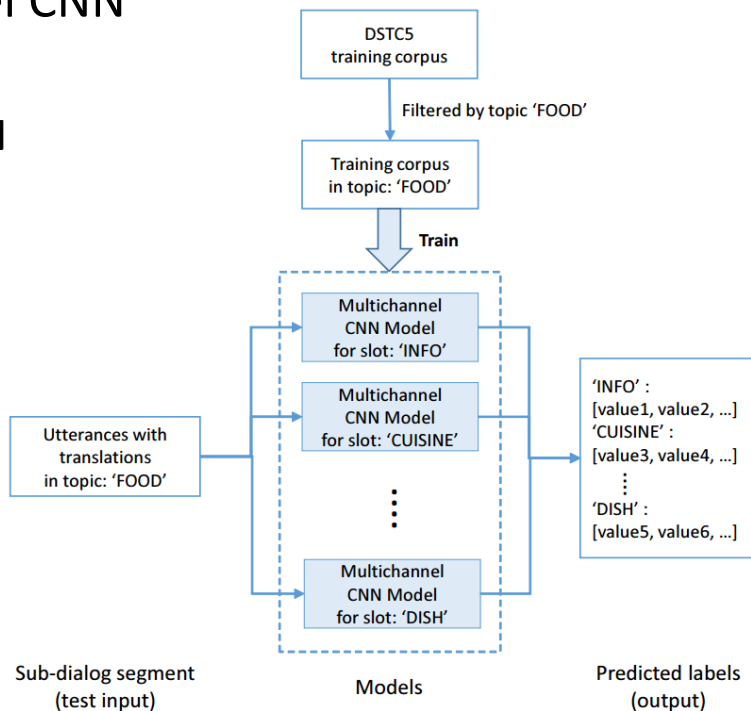
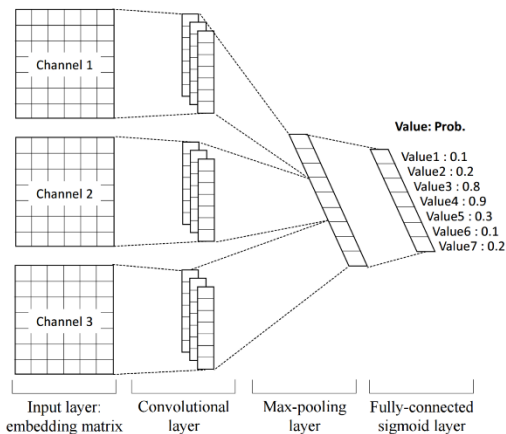
Multichannel Tracker (Shi et al., 2016)

60

<https://arxiv.org/abs/1701.06247>

□ Training a multichannel CNN for each slot

- ▣ Chinese character CNN
- ▣ Chinese word CNN
- ▣ English word CNN



DST Evaluation

61

□ Metric

- ▣ Tracked state accuracy with respect to user goal
- ▣ L2-norm of the hypothesized dist. and the true label

$$L_2 = (1 - p_i)^2 + \sum_{j \neq i} p_j^2$$

- ▣ Recall/Precision/F-measure individual slots

Dialog State Tracking Challenge (DSTC)

(Williams et al. 2013, Henderson et al. 2014, Henderson et al. 2014, Kim et al. 2016, Kim et al. 2016)

62

| Challenge | Type | Domain | Data Provider | Main Theme |
|------------------------------|---------------|---------------------|---------------|---------------------|
| <u>DSTC1</u> | Human-Machine | Bus Route | CMU | Evaluation Metrics |
| <u>DSTC2</u> | Human-Machine | Restaurant | U. Cambridge | User Goal Changes |
| <u>DSTC3</u> | Human-Machine | Tourist Information | U. Cambridge | Domain Adaptation |
| <u>DSTC4</u> | Human-Human | Tourist Information | I2R | Human Conversation |
| <u>DSTC5</u> | Human-Human | Tourist Information | I2R | Language Adaptation |

DSTC1

63

□ Type: Human-Machine

□ Domain: Bus Route

System: East Pittsburgh Bus Schedules. Say a bus route, like 28X, or say I'm not sure.
hello(), request(route), example(route=28x), example(route=dont_know)

User: 61A

SLU: 0.77 inform(route=61a)
0.12 inform(route=61)
0.01 inform(route=61d)

System: Okay, 61A. To change, say go back. Where are you leaving from?
impl-conf(route=61a), example(act=goback), request(from)

User: Downtown

SLU: 0.59 inform(from.desc=downtown)
0.10 inform(from.desc=from downtown)

System: Okay, downtown. You can always say go back. And where are you going to?
impl-conf(from.desc=downtown), example(act=goback), request(to)

User: East Pittsburgh East Pittsburgh

SLU: 0.25 inform(to.desc=pittsburgh)
0.20 inform(to.desc=east pittsburgh)
0.01 inform(to.desc=to pittsburgh)

System: Pittsburgh. Is that right?
expl-conf(to.desc=pittsburgh)

User: No

SLU: 0.71 negate()
0.07 affirm()

System: Oh, sorry, where are you going to?
sorry(), request(to)

User: East Pittsburgh

SLU: 0.97 inform(to.desc=east pittsburgh)

System: Okay, East Pittsburgh. Do you want times for the next few buses? Say yes or no.
impl-confirm(to.desc=east pittsburgh), expl-confirm(act=want-next-bus),
example(act=affirm), example(act=negate)

User: Yes

SLU: 0.97 affirm()

System: Ok, the next few buses. To change, say go back. At 7:45 PM today, there is a 61 A, from Fifth Avenue And Market Street, Downtown, arriving Dynamo Alley And Beech Avenue, In East Pittsburgh, at 8:42 PM. Say repeat, next, previous, or start over.
impl-conf(act=want-next-bus), example(act=goback),
schedule(result.from.time.hour=7, result.from.time.minute=45,
result.from.time.ampm=pm, result.from.date.day=today, result.route=61a,
result.from.desc=fifth avenue and market street, downtown, result.to.desc=dynamo
alley and beech avenue, in east pittsburgh, result.to.time.hour=8,
result.to.time.minute=42, result.to.time.ampm=pm), example(act=repeat),
example(act=nextbus), example(act=prevbus), example(act=restart)

User: [hang up]

DSTC4-5

64

□ Type: Human-Human

□ Domain: Tourist Information

{Topic: Accommodation; Type: Hostel; Pricerange: Cheap; GuideAct: ACK; TouristAct: REQ}

Tourist: Can you give me some uh- tell me some cheap rate hotels, because I'm planning just to leave my bags there and go somewhere take some pictures.

Guide: Okay. I'm going to recommend firstly you want to have a backpack type of hotel, right?

Tourist: Yes. I'm just gonna bring my backpack and my buddy with me. So I'm kinda looking for a hotel that is not that expensive. Just gonna leave our things there and, you know, stay out the whole day.

Guide: Okay. Let me get you hm hm. So you don't mind if it's a bit uh not so roomy like hotel because you just back to sleep.

Tourist: Yes. Yes. As we just gonna put our things there and then go out to take some pictures.

Guide: Okay, um-

Tourist: Hm.

{Topic: Accommodation; NAME: InnCrowd Backpackers Hostel; GuideAct: REC; TouristAct: ACK}

Guide: Let's try this one, okay?

Tourist: Okay.

Guide: It's InnCrowd Backpackers Hostel in Singapore. If you take a dorm bed per person only twenty dollars. If you take a room, it's two single beds at fifty nine dollars.

Tourist: Um. Wow, that's good.

Guide: Yah, the prices are based on per person per bed or dorm. But this one is room. So it should be fifty nine for the two room. So you're actually paying about ten dollars more per person only.

Tourist: Oh okay. That's- the price is reasonable actually. It's good.

Concluding Remarks

65

- **Dialogue state tracking (DST)** of DM has Markov assumption to model the user goal and be robust to errors
- Generative models for DST are based on POMDP
 - Hidden Information State (HIS)
 - state \rightarrow user goal, user action dialogue history
 - transitions are hand-crafted and the goals are grouped together to allow tractable belief tracking
 - Bayesian Update of Dialogue State (BUDS)
 - further factorizes the state
 - allows tractable belief tracking and learning of the shapers of distributions via expectation propagation
- Discriminative models directly estimate dialogue states given arbitrary input features

