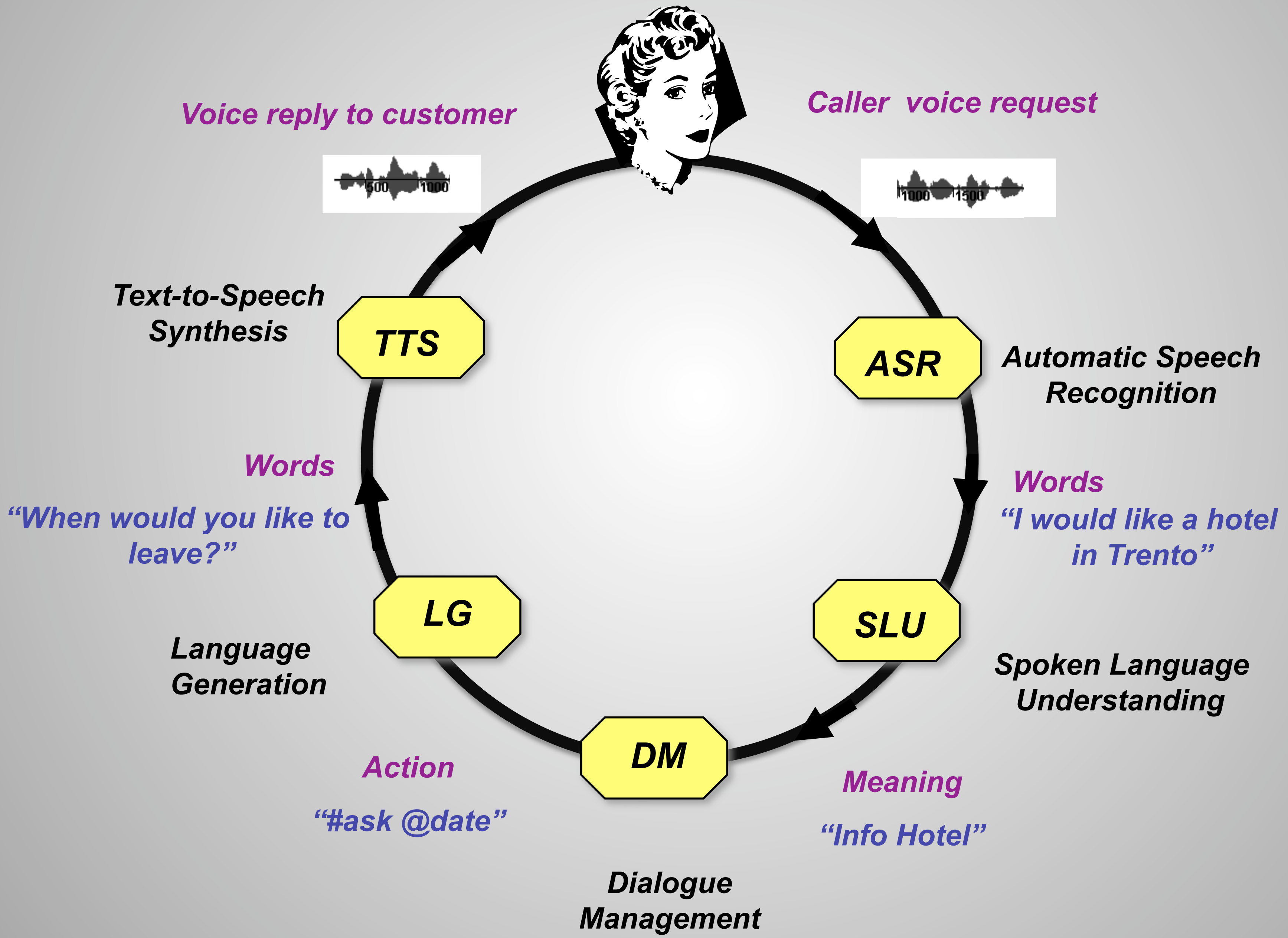
Spoken Dialog Systems: from Rule-Based Systems to Markov Decision Processes

S. Varges
G. Riccardi
S. Quarteroni
A. Ivanov
P. Roberti
AMI^2 *Lab, casa.disi.unitn.it*
EECS Department, University of Trento, Italy
{varges,riccardi,silviaq,ivanov,roberti}@disi.unitn.it

Outline

- Motivation and Problem
- Spoken Dialog Systems
- Learning Dialog Models
- Markov Decision Processes
- Adaptive SDS Demo

Spoken Dialog Systems



Applications

- Call Center Automation
  - Customer Care
  - Business Information services (e.g. yellow pages)
- Spoken/Multimodal Access to web documents and services (e.g. voice portals)
  - Smart Phones
- Interactive in-car navigational systems
- Personal Agents
  - Butler agents, Shopping assistants
  - Assistants for elderly people or people with disabilities
- Consumer Robotics

Dialog Example

System : Hi, This is JD Travel Agent, How May I Help You?
Caller : Hi, I need a four stars hotel in Trento
ASR : I need a for stars hotel in
SLU : INFO(0.8) HOTEL(0.7)
DM : Request.Date.CheckIn
Turn 1

System : When do you want to check in?
Caller : I plan to arrive on um May 30
ASR : I arrive on May 30
SLU : DATE(0.8)
DM : Request.DateCheckOut
Turn 2

System : When do you want to check out?
Caller : June 6
ASR : June
SLU : DATE(0.7)
DM : Request.DateCheckOut
Turn 3

System : Please repeate the checkout date?
Caller : June 6
ASR : June 6
SLU : DATE(0.9)
DM : Confirm.CheckIn,Checkout
Turn 4

time

Motivation

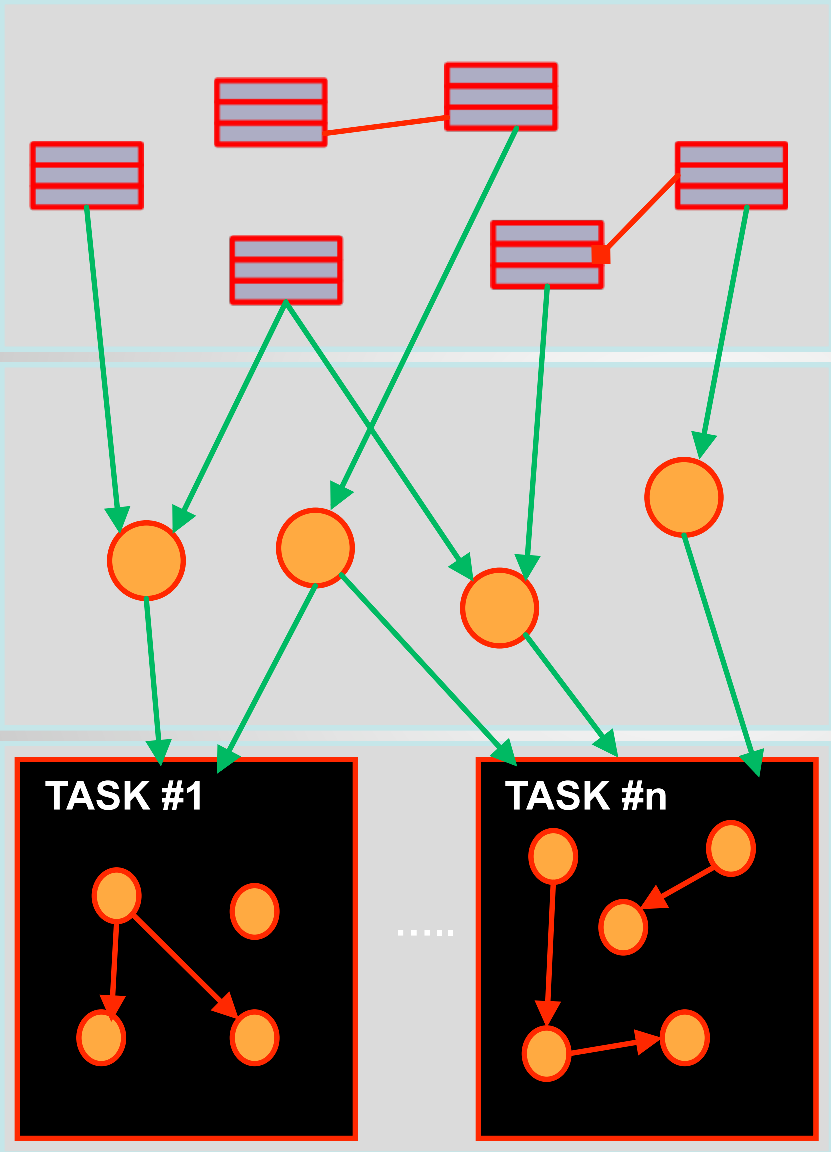
- How to automatically train SDS such that:
  - Minimize the amount of time and resources (human and data)
  - Maximize the effectiveness of SDS (e.g. task completion rates)
    - wrt to system performance (e.g. ASR and Language Understanding errors)
    - wrt to user input and behavior variability (e.g. hang-ups, language etc.)

Rule-Based Systems

CONCEPT ONTOLOGY (XML)

ACTION ONTOLOGY (XML)

TASK PLANNING (XML)



1) Domain Representation

2) Task Representation

3) Task Execution

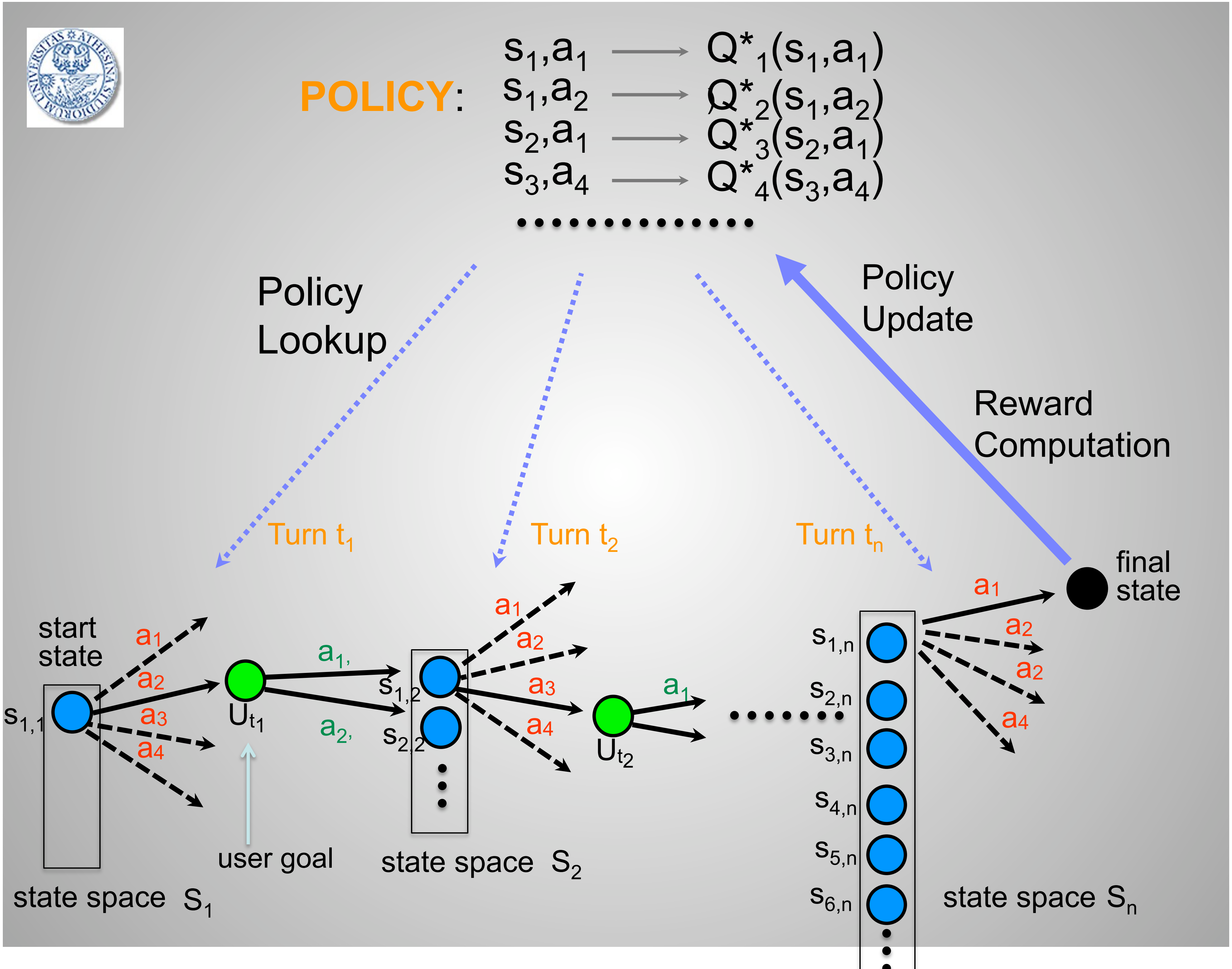
are hand-coded

Markov Decision Processes

- MDP: dialog as sequential decision process with transitional uncertainty
  - Maintains one dialogue state *s* at each time *t*
  - action a_t chosen w.r.t state *s*
- POMDP: adds observational uncertainty (ASR, SLU, …)
  - maintains large number of `parallel' dialogue states: the `belief'
  - action a_t chosen w.r.t the distribution of states
  - Training: by RL, with user simulations

State Representation
an example from tourist domain

Concept	Value	Confidence	Rank	Recency	Verification Status
Activity	(n tasks)	0.0 -1.0	1, 2	1, 0	-
Location	(m)				positive
StarRating	1-5				negative
Month_start	1-12				
Day_start	1-31				
Month_end	1-12				
Day_end	1-31				
Duration	1-90				
Quit-user	1, 0				
Operator-user	1, 0				
………….	…	….	….	…..	…..



POLICY:
$s_1,a_1 \longrightarrow Q^*_1(s_1,a_1)$
$s_1,a_2 \longrightarrow Q^*_2(s_1,a_2)$
$s_2,a_1 \longrightarrow Q^*_3(s_2,a_1)$
$s_3,a_4 \longrightarrow Q^*_4(s_3,a_4)$
Policy Lookup
Policy Update
Reward Computation
Turn t_1
Turn t_2
Turn t_n
start state
$s_{1,1}$
a_1 a_2 a_3 a_4
U_{t_1}
user goal
$a_{1,}$
$a_{2,}$
$s_{1,2}$
$s_{2,2}$
U_{t_2}
state space S_1
state space S_2
$s_{1,n}$ $s_{2,n}$ $s_{3,n}$ $s_{4,n}$ $s_{5,n}$ $s_{6,n}$
final state
state space S_n

Legend

a_u	user act at time t
a_s	system action at time t
S_t	system state space at time t
$s_{r,t}$	system state of rank r at time t
U_{t1}	user state at turn n
s_k,a_m	(Policy): action m at state k
Q^*_k	(Policy): value of s_k,a_m:

Reward function

Reward $R = w_1 M - w_2 N - w_3 D - w_4 E$

where

M: #matches user goal concepts – system concepts

N: #mismatches incl. unknown

D: duration in turns

E: ending cost

Factors and Weights used in demo system:

$w_{1:}$ 10, $w_{3:}$ 0.25, all other weights =1

Ending costs E: operator 10, hangup/quit 20, DB 5

Value update in policy

(Following Levin et al. 2000:)

n= number of sessions

C= cost

Q*= estimate of optimal state-action value

$$Q_t^*(s',a') = C(s',a') / n + Q_{t-1}^* \times (n-1)/n$$

Exploration vs Exploitation

- Current dialog systems do not explore, rather exploit hardwired and expensive heuristic strategies.
- Conversational Agent needs to find trade-off between exploration and exploitation reward
- Most natural (wrt cognitive process) strategy

Adaptive Learning

- Action selection strategy
  - Softmax (τ): actions selected according estimated probability distribution (e.g. Gibbs Distribution)

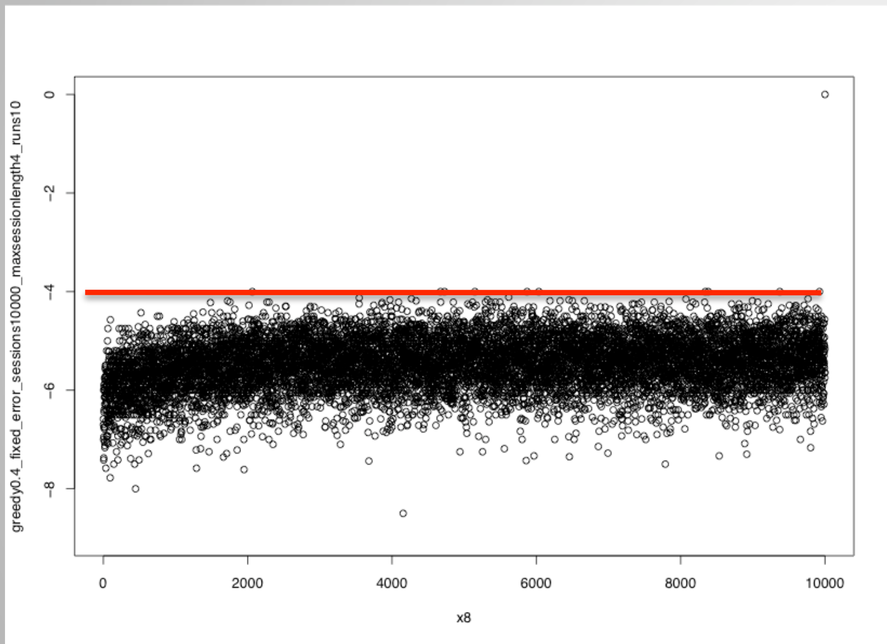
$$\frac{e^{Q_t(a)/t}}{\sum_b e^{Q_t(b)/t}}$$

  - Greedy (ε): exploitation is selected with prob ε and exploration with prob (1-ε).
- Example
  - Adaptive Spoken Dialog System seeking to acquire two attribute slots (day and month)

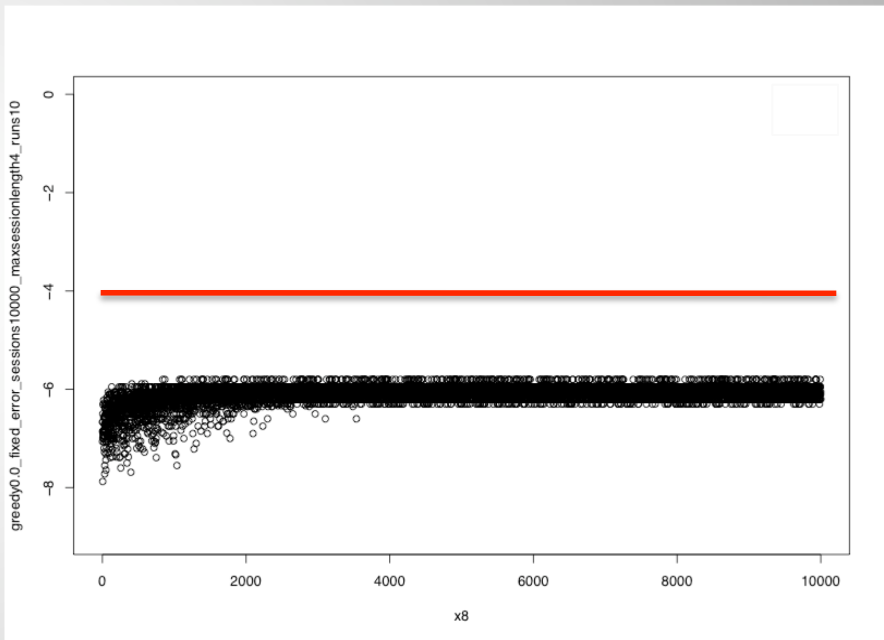
Exploration vs Exploitation

Simulations

40% exploration, 60% exploitation
Optimal Reward = -4



0% exploration, 100% exploitation:
Does not find optimal dialogue strategy



Conclusion

- **Human-Machine Interaction**

- **Learning Systems based on**
  - **Human feedback**
  - **Uncertainty user/world state**
  - **Reward structure**
  - **Adaptive strategy computation**