Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states $S$ the agent can be in
  - A set of actions $A$ the agent can take
  - A goal $G$, which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy $\pi$ for what action to take in any particular state.
Goals are not enough

- Goal: user satisfaction
- OK, that’s all very well, but
  - Many things influence user satisfaction
  - We don’t know user satisfaction til after the dialogue is done
  - How do we know, state by state and action by action, what the agent should do?
- We need a more helpful metric that can apply to each state

Utility

- A utility function
  - maps a state or state sequence
  - onto a real number
  - describing the goodness of that state
  - I.e. the resulting “happiness” of the agent
- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent’s expected utility
Markov Decision Processes

- Or MDP
- Characterized by:
  - a set of states $S$ an agent can be in
  - a set of actions $A$ the agent can take
  - A reward $r(a,s)$ that the agent receives for taking an action in a state

  (+ Some other things I’ll come back to (gamma, state transition probabilities))

A brief tutorial example

- A Day-and-Month dialogue system
- Goal: fill in a two-slot frame:
  - Month: November
  - Day: 12th
- Via the shortest possible interaction with user
What is a state?

- In principle, MDP state could include any possible information about dialogue
  - Complete dialogue history so far
- Usually use a much more limited set
  - Values of slots in current frame
  - Most recent question asked to user
  - Users most recent answer
  - ASR confidence
  - etc

State in the Day-and-Month example

- Values of the two slots day and month.
- Total:
  - 2 special initial state $s_i$ and $s_f$.
  - 365 states with a day and month
  - 1 state for leap year
  - 12 states with a month but no day
  - 31 states with a day but no month
  - 411 total states
Actions in MDP models of dialogue

- Speech acts!
  - Ask a question
  - Explicit confirmation
  - Rejection
  - Give the user some database information
  - Tell the user their choices
- Do a database query

Actions in the Day-and-Month example

- \( a_d \): a question asking for the day
- \( a_m \): a question asking for the month
- \( a_{dm} \): a question asking for the day+month
- \( a_f \): a final action submitting the form and terminating the dialogue
A simple reward function

- For this example, let's use a cost function
- A cost function for entire dialogue
- Let
  - \(N_i=\) number of interactions (duration of dialogue)
  - \(N_e=\) number of errors in the obtained values (0-2)
  - \(N_f=\) expected distance from goal
    - (0 for complete date, 1 if either data or month are missing, 2 if both missing)
- Then (weighted) cost is:
- \[C = w_i\times N_i + w_e\times N_e + w_f\times N_f\]

2 possible policies

Policy 1 (directive)

- \(d=0\) \(m=0\) Which day?
- \(d=D\) \(m=0\) Which month?
- \(d=D\) \(m=M\)  Goodbye.
- \(d=1\) \(m=1\)

\[c_1 = -3w_i + 2p_d w_e\]

Policy 2 (open)

- \(d=0\) \(m=0\) What date?
- \(d=D\) \(m=M\) Goodbye.
- \(d=1\) \(m=1\)

\[c_2 = -2w_i + 2p_o w_e\]

\(P_o=\) probability of error in directive prompt

\(P_o=\) probability of error in open prompt
2 possible policies

Strategy 1 is better than strategy 2 when improved error rate justifies longer interaction:

\[ p_o - p_d > \frac{w_i}{2w_e} \]

Policy 1 (directive)

- \( d=0 \) \( m=0 \) \( \text{Which day?} \)
- \( d=D \) \( m=0 \) \( \text{Which month?} \)
- \( d=D \) \( m=M \) \( \text{Goodbye.} \)
- \( d=1 \) \( m=-1 \)

\( c_1 = -3w_i + 2p_d w_e \)

Policy 2 (open)

- \( d=0 \) \( m=0 \) \( \text{What date?} \)
- \( d=D \) \( m=M \) \( \text{Goodbye.} \)
- \( d=1 \) \( m=-1 \)

\( c_2 = -2w_i + 2p_o w_e \)

That was an easy optimization

- Only two actions, only tiny \# of policies
- In general, number of actions, states, policies is quite large
- So finding optimal policy \( \pi^* \) is harder
- We need reinforcement learning
- Back to MDPs:
### MDP

- We can think of a dialogue as a trajectory in state space.
- The best policy $\pi^*$ is the one with the greatest expected reward over all trajectories.
- How to compute a reward for a state sequence?

$$S_1 \rightarrow a_1, r_1 \quad S_2 \rightarrow a_2, r_2 \quad S_3 \rightarrow a_3, r_3 \quad \cdots$$

### Reward for a state sequence

- One common approach: discounted rewards.
- Cumulative reward $Q$ of a sequence is discounted sum of utilities of individual states.

$$Q([s_0, a_0, s_1, a_1, s_2, a_2, \ldots]) = R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots$$

- Discount factor $\gamma$ between 0 and 1.
- Makes agent care more about current than future rewards; the more future a reward, the more discounted its value.
The Markov assumption

- MDP assumes that state transitions are Markovian

\[ P(s_{t+1} \mid s_t, s_{t-1}, \ldots, s_o, a_t, a_{t-1}, \ldots, a_o) = P_T(s_{t+1} \mid s_t, a_t) \]

Expected reward for an action

- Expected cumulative reward \( Q(s, a) \) for taking a particular action from a particular state can be computed by Bellman equation:

\[ Q(s, a) = R(s, a) + \gamma \sum_{s'} P(s' \mid s, a) \max_{a'} Q(s', a') \]

- Expected cumulative reward for a given state/action pair is:
  - immediate reward for current state
  - + expected discounted utility of all possible next states \( s' \)
  - Weighted by probability of moving to that state \( s' \)
  - And assuming once there we take optimal action \( a' \)
What we need for Bellman equation

- A model of $p(s'|s,a)$
- Estimate of $R(s,a)$
- How to get these?
- If we had labeled training data
  - $p(s'|s,a) = C(s,s',a)/C(s,a)$
- If we knew the final reward for whole dialogue $R(s_1,a_1,s_2,a_2,...,s_n)$
- Given these parameters, can use value iteration algorithm to learn Q values (pushing back reward values over state sequences) and hence best policy

Final reward

- What is the final reward for whole dialogue $R(s_1,a_1,s_2,a_2,...,s_n)$?
- This is what our automatic evaluation metric PARADISE computes!
- The general goodness of a whole dialogue!!!
How to estimate $p(s'|s,a)$ without labeled data

- Have random conversations with real people
  - Carefully hand-tune small number of states and policies
  - Then can build a dialogue system which explores state space by generating a few hundred random conversations with real humans
  - Set probabilities from this corpus
- Have random conversations with simulated people
  - Now you can have millions of conversations with simulated people
  - So you can have a slightly larger state space

An example


- NJFun system, people asked questions about recreational activities in New Jersey
- Idea of paper: use reinforcement learning to make a small set of optimal policy decisions
Very small # of states and acts

- **States**: specified by values of 8 features
  - Which slot in frame is being worked on (1-4)
  - ASR confidence value (0-5)
  - How many times a current slot question had been asked
  - Restrictive vs. non-restrictive grammar
  - Result: 62 states

- **Actions**: each state only 2 possible actions
  - Asking questions: System versus user initiative
  - Receiving answers: explicit versus no confirmation.

Ran system with real users

- 311 conversations
- Simple binary reward function
  - 1 if competed task (finding museums, theater, winetasting in NJ area)
  - 0 if not
- System learned good dialogue strategy: Roughly
  - Start with user initiative
  - Backoff to mixed or system initiative when re-asking for an attribute
  - Confirm only a lower confidence values
State of the art

- Only a few such systems
  - From (former) ATT Laboratories researchers, now dispersed
  - And Cambridge UK lab
- Hot topics:
  - Partially observable MDPs (POMDPs)
  - We don’t REALLY know the user’s state (we only know what we THOUGHT the user said)
  - So need to take actions based on our BELIEF, i.e. a probability distribution over states rather than the “true state”

Summary

- Utility-based conversational agents
  - Policy/strategy for:
    - Confirmation
    - Rejection
    - Open/directive prompts
    - Initiative
    - +?????
  - MDP
  - POMDP
Summary

- The Linguistics of Conversation
- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Evaluation
- Utility-based conversational agents
  - MDP, POMDP