Lecture 10: Information State and Utility-based dialogue systems
Outline

- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Information-State
    - Dialogue-Act Detection
    - Dialogue-Act Generation
- Utility-based conversational agents
  - MDP, POMDP
- Evaluation
Information-State and Dialogue Acts

- For more than just form-filling
- Need to:
  - Decide when the user has asked a question, made a proposal, rejected a suggestion
  - Ground a user’s utterance, ask clarification questions, suggest plans
- Need models of interpretation and generation
  - Speech acts and grounding
  - More sophisticated representation of dialogue context than just a list of slots
Information-state architecture

- Information state
- Dialogue act interpreter
- Dialogue act generator
- Set of update rules
  - Update dialogue state as acts are interpreted
  - Generate dialogue acts
- Control structure to select which update rules to apply
Dialog acts

- Also called “conversational moves”
- An act with (internal) structure related specifically to its dialogue function
- Incorporates ideas of grounding
- Incorporates other dialogue and conversational functions that Austin and Searle didn’t seem interested in
Verbmobil task

- Two-party scheduling dialogues
- Speakers were asked to plan a meeting at some future date
- Data used to design conversational agents which would help with this task
- (cross-language, translating, scheduling assistant)
<table>
<thead>
<tr>
<th>Verbmobil Dialogue Acts</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THANK</strong></td>
<td>thanks</td>
</tr>
<tr>
<td><strong>GREET</strong></td>
<td>Hello Dan</td>
</tr>
<tr>
<td><strong>INTRODUCE</strong></td>
<td>It’s me again</td>
</tr>
<tr>
<td><strong>BYE</strong></td>
<td>Alright, bye</td>
</tr>
<tr>
<td><strong>REQUEST-COMMENT</strong></td>
<td>How does that look?</td>
</tr>
<tr>
<td><strong>SUGGEST</strong></td>
<td>June 13th through 17th</td>
</tr>
<tr>
<td><strong>REJECT</strong></td>
<td>No, Friday I’m booked all day</td>
</tr>
<tr>
<td><strong>ACCEPT</strong></td>
<td>Saturday sounds fine</td>
</tr>
<tr>
<td><strong>REQUEST-SUGGEST</strong></td>
<td>What is a good day of the week for you?</td>
</tr>
<tr>
<td><strong>INIT</strong></td>
<td>I wanted to make an appointment with you</td>
</tr>
<tr>
<td><strong>GIVE_REASON</strong></td>
<td>Because I have meetings all afternoon</td>
</tr>
<tr>
<td><strong>FEEDBACK</strong></td>
<td>Okay</td>
</tr>
<tr>
<td><strong>DELIBERATE</strong></td>
<td>Let me check my calendar here</td>
</tr>
<tr>
<td><strong>CONFIRM</strong></td>
<td>Okay, that would be wonderful</td>
</tr>
<tr>
<td><strong>CLARIFY</strong></td>
<td>Okay, do you mean Tuesday the 23rd?</td>
</tr>
<tr>
<td>Category</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STATEMENT</td>
<td>a claim made by the speaker</td>
</tr>
<tr>
<td>INFO-REQUEST</td>
<td>a question by the speaker</td>
</tr>
<tr>
<td>CHECK</td>
<td>a question for confirming information</td>
</tr>
<tr>
<td>INFLUENCE-ON-ADDRESSEE (=Searle's directives)</td>
<td></td>
</tr>
<tr>
<td>OPEN-OPTION</td>
<td>a weak suggestion or listing of options</td>
</tr>
<tr>
<td>ACTION-DIRECTIVE</td>
<td>an actual command</td>
</tr>
<tr>
<td>INFLUENCE-ON-SPEAKER (=Austin's commissives)</td>
<td></td>
</tr>
<tr>
<td>OFFER</td>
<td>speaker offers to do something</td>
</tr>
<tr>
<td>COMMIT</td>
<td>speaker is committed to doing something</td>
</tr>
<tr>
<td>CONVENTIONAL</td>
<td>other</td>
</tr>
<tr>
<td>OPENING</td>
<td>greetings</td>
</tr>
<tr>
<td>CLOSING</td>
<td>farewells</td>
</tr>
<tr>
<td>THANKING</td>
<td>thanking and responding to thanks</td>
</tr>
</tbody>
</table>
DAMSL: backward looking function

AGREEMENT speaker's response to previous proposal
ACCEPT accepting the proposal
ACCEPT-PART accepting some part of the proposal
MAYBE neither accepting nor rejecting the proposal
REJECT-PART rejecting some part of the proposal
REJECT rejecting the proposal
HOLD putting off response, usually via subdialogue
ANSWER answering a question
UNDERSTANDING whether speaker understood previous
SIGNAL-NON-UNDER. speaker didn't understand
SIGNAL-UNDER. speaker did understand
ACK demonstrated via continuers or assessment
REPEAT-REPHRASE demonstrated via repetition or reformulation
COMPLETION demonstrated via collaborative completion
A DAMSL Labeling

[info-req,ack]    A1:    And, what day in May did you want to travel?
[assert, answer]  C2:    OK uh I need to be there for a meeting that’s from the 12th to the 15th.
[info-req,ack]    A2:    And you’re flying into what city?
[assert, answer]  C3:    Seattle.
[info-req,ack]    A3:    And what time would you like to leave Pittsburgh?
[check, hold]     C4:    Uh hmm I don’t think there’s many options for non-stop.
[assert]          The three non-stops today.
[info-req]        C5:    What are they?
[assert, open-    A5:    The first one departs PGH at 10:00am arrives Seattle at 12:05 their time. The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
option]           
[accept, ack]     C6:    OK I’ll take the 5ish flight on the night before on the 11th.
[check, ack]      A6:    On the 11th?
[assert, ack]     OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
### Conversation Acts

Traum and Hinkelman (1992)

<table>
<thead>
<tr>
<th>Act Type</th>
<th>Sample Acts</th>
</tr>
</thead>
<tbody>
<tr>
<td>turn-taking</td>
<td>take-turn, keep-turn, release-turn, assign-turn</td>
</tr>
<tr>
<td>grounding</td>
<td>acknowledge, repair, continue</td>
</tr>
<tr>
<td>core speech acts</td>
<td>inform, wh-question, accept, request, offer</td>
</tr>
<tr>
<td>argumentation</td>
<td>elaborate, summarize, question-answer, clarify</td>
</tr>
</tbody>
</table>
Generating Dialogue Acts

- Two examples
  - Confirmation
  - Rejection
Confirmation

- Errors: Speech is a pretty errorful channel
  - Humans use grounding to confirm that they heard correctly
  - ASR is way worse than humans!
- Dialog systems need to do even more grounding and confirmation than humans
  - Users are confused when system doesn’t give explicit acknowledgment signal.

Stifelman et al. (1993), Yankelovich et al. (1995)
Explicit confirmation

S: Which city do you want to leave from?
U: Baltimore
S: Do you want to leave from Baltimore?
U: Yes
Explicit confirmation

U: I’d like to fly from Denver Colorado to New York City on September 21st in the morning on United Airlines

S: Let’s see then. I have you going from Denver Colorado to New York on September 21st. Is that correct?

U: Yes
Implicit confirmation: display

U: I’d like to travel to Berlin
S: When do you want to travel to Berlin?

U: Hi I’d like to fly to Seattle Tuesday morning
S: Traveling to Seattle on Tuesday, August eleventh in the morning. Your name?
Implicit vs. Explicit

- Complementary strengths
- Explicit: easier for users to correct systems’ mistakes (can just say “no”)
- But explicit is cumbersome and long
- Implicit: much more natural, quicker, simpler (if system guesses right).
Implicit and Explicit

- Early systems: all-implicit or all-explicit
- Modern systems: adaptive
- How to decide?
  - ASR system can give confidence metric.
  - This expresses how convinced system is of its transcription of the speech
  - If high confidence, use implicit confirmation
  - If low confidence, use explicit confirmation
Computing confidence

- Simplest: use acoustic log-likelihood of user’s utterance
- More features
  - Prosodic: utterances with longer pauses, F0 excursions, longer durations
  - Backoff: did we have to backoff in the LM?
  - Cost of an error: Explicit confirmation before moving money or booking flights
Rejection

- “I’m sorry, I didn’t understand that.”
- Reject when:
  - ASR confidence is low
  - Best interpretation is semantically ill-formed
- Might have four-tiered level of confidence:
  - Below confidence threshold, reject
  - Above threshold, explicit confirmation
  - If even higher, implicit confirmation
  - Even higher, no confirmation
Automatic Interpretation of Dialogue Acts

- How do we automatically identify dialogue acts?
  - Given an utterance:
    - Decide whether it is a QUESTION, STATEMENT, SUGGEST, or ACK
  - Perhaps we can just look at the form of the utterance to decide?
Can we just use the surface syntactic form?

YES-NO-Qs have auxiliary-before-subject syntax:

Will breakfast be served on USAir 1557?

STATEMENTS have declarative syntax:

I don’t care about lunch

COMMANDs have imperative syntax:

Show me flights from Milwaukee to Orlando on Thursday night
<table>
<thead>
<tr>
<th></th>
<th>Surface form</th>
<th>Speech act</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can I have the rest of your sandwich?</td>
<td>Question</td>
<td>Request</td>
</tr>
<tr>
<td>I want the rest of your sandwich</td>
<td>Declarative</td>
<td>Request</td>
</tr>
<tr>
<td>Give me your sandwich!</td>
<td>Imperative</td>
<td>Request</td>
</tr>
</tbody>
</table>
Dialogue Act ambiguity

Can you give me a list of the flights from Atlanta to Boston?

- This looks like an INFO-REQUEST.
- If so, the answer is:
  - YES.
- But really it’s a DIRECTIVE or REQUEST, a polite form of:
  Please give me a list of the flights...
- What looks like a QUESTION can be a REQUEST
Dialogue Act ambiguity

- Similarly, what looks like a STATEMENT can be a QUESTION:

<table>
<thead>
<tr>
<th>Us</th>
<th>OPEN-OPTION</th>
<th>I was wanting to make some arrangements for a trip that I’m going to be taking uh to LA uh beginnnning of the week after next</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ag</td>
<td>HOLD</td>
<td>OK uh let me pull up your profile and I’ll be right with you here. [pause]</td>
</tr>
<tr>
<td>Ag</td>
<td>CHECK</td>
<td>And you said you wanted to travel next week?</td>
</tr>
<tr>
<td>Us</td>
<td>ACCEPT</td>
<td>Uh yes.</td>
</tr>
</tbody>
</table>
Indirect speech acts

Utterances which use a surface statement to ask a question
Utterances which use a surface question to issue a request
DA interpretation as statistical classification: Features

- **Words and Collocations:**
  - *Please or would you*: good cue for REQUEST
  - *Are you*: good cue for INFO-REQUEST

- **Prosody:**
  - *Rising pitch* is a good cue for INFO-REQUEST
  - *Loudness/stress* can help distinguish yeah/AGREEMENT from yeah/BACKCHANNEL

- **Conversational Structure**
  - *Yeah* following a proposal is probably AGREEMENT; *yeah* following an INFORM probably a BACKCHANNEL
An example of dialogue act detection: Correction Detection

- If system misrecognizes an utterance, and either
  - Rejects
  - Via confirmation, displays its misunderstanding
- Then user has a chance to make a
  correction
    - Repeat themselves
    - Rephrasing
    - Saying “no” to the confirmation question.
Corrections

- Unfortunately, corrections are harder to recognize than normal sentences!
- Swerts et al. (2000): corrections misrecognized twice as often (in terms of WER) as non-corrections!!!
- Why?
  - Prosody seems to be largest factor: **hyperarticulation**
  - Liz Shriberg example:
    - “NO, I am DE-PAR-TING from Jacksonville”
  - Bettina Braun example from a talking elevator
    - “In den VIERTEN Stock”
<table>
<thead>
<tr>
<th>Turn</th>
<th>Corr Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>On which day of the week do you want to leave?</td>
</tr>
<tr>
<td>S</td>
<td>Sorry, I didn’t hear anything from you. Please tell me your departure date</td>
</tr>
<tr>
<td>U</td>
<td><em>Um tomorrow</em></td>
</tr>
<tr>
<td>S</td>
<td>You can say the departure date. For example, you can say ’today’ or ’tomorrow’. Or you can say ’Monday’, ’Tuesday’, ’Wednesday’...</td>
</tr>
<tr>
<td>U</td>
<td><em>Tomorrow</em></td>
</tr>
<tr>
<td>S</td>
<td>You can say the departure date. For example, you can say ’today’ or ’tomorrow’. Or you can say ’Monday’, ’Tuesday’, ’Wednesday’...</td>
</tr>
<tr>
<td>U</td>
<td><em>Saturday</em></td>
</tr>
<tr>
<td>S</td>
<td>Do you want to leave on Saturday?</td>
</tr>
<tr>
<td>U</td>
<td>Yes</td>
</tr>
<tr>
<td>S</td>
<td>What time do you want to leave?</td>
</tr>
<tr>
<td>U</td>
<td>Eight o’clock A M</td>
</tr>
<tr>
<td>S</td>
<td>Do you want to leave around ten o’clock p-m</td>
</tr>
<tr>
<td>U</td>
<td>Eight o’clock</td>
</tr>
<tr>
<td>S</td>
<td>Do you want to leave around ten o’clock p-m</td>
</tr>
<tr>
<td>U</td>
<td>No</td>
</tr>
<tr>
<td>S</td>
<td>What time do you want to leave?</td>
</tr>
<tr>
<td>U</td>
<td>Eight o’clock A M</td>
</tr>
</tbody>
</table>
Machine learning to detect user corrections: features

- Lexical information (no, correction, I don’t, swear words)
- Prosodic indicators of hyperarticulation
  - increases in F0 range, pause duration, word duration
- Length
- ASR confidence
- LM probability
- Various dialogue features (repetition)
Prosodic Features

- Shriberg et al. (1998)
- Decision tree trained on simple acoustically-based prosodic features
  - Slope of F0 at the end of the utterance
  - Average energy at different places in utterance
  - Various duration measures
  - All normalized in various ways
- These helped distinguish
  - Statement (S)
  - Yes-no-question (QY)
  - Declarative question (QD) (“You’re going to the store?”)
  - Wh-question (QW)
Prosodic Decision Tree for making S/QY/QW/QD decision
Dialogue System Evaluation

• Always two kinds of evaluation
  • Extrinsic: embedded in some external task
  • Intrinsic: evaluating the component as such

• What constitutes success or failure for a dialogue system?
Reasons for Dialogue System Evaluation

1. A metric to compare systems
   • can’t improve it if we don’t know where it fails
   • can’t decide between two systems without a goodness metric

2. A metric as an input to reinforcement learning:
   • automatically improve conversational agent performance via learning
PARADISE evaluation

- Maximize Task Success
- Minimize Costs
  - Efficiency Measures
  - Quality Measures
Task Success

- % of subtasks completed
- Correctness of each questions/answer/error msg
- Correctness of total solution
  - Error rate in final slots
    - Generalization of Slot Error Rate
- Users’ perception of whether task was completed
Efficiency Cost

Polifroni et al. (1992), Danieli and Gerbino (1995)
Hirschman and Pao (1993)

- Total elapsed time in seconds or turns
- Number of queries
- Turn correction ration: number of system or user turns used solely to correct errors, divided by total number of turns
Quality Cost

- # of times ASR system failed to return any sentence
- # of ASR rejection prompts
- # of times user had to barge-in
- # of time-out prompts
- Inappropriateness (verbose, ambiguous) of system’s questions, answers, error messages
Concept accuracy:

- “Concept accuracy” or “Concept error rate”
- % of semantic concepts that the NLU component returns correctly
- I want to arrive in Austin at 5:00
  - DESTCITY: Boston
  - Time: 5:00
- Concept accuracy = 50%
- Average this across entire dialogue
- “How many of the sentences did the system understand correctly”
- Can be used as either quality cost or task success
PARADISE: Regress against user satisfaction

- Maximize user satisfaction
  - Maximize task success
  - Minimize costs
    - Efficiency measures
    - Quality measures
Regressing against user satisfaction

- Questionnaire to assign each dialogue a "user satisfaction rating": this is dependent measure
- Set of cost and success factors are independent measures
- Use regression to train weights for each factor
Experimental Procedures

- Subjects given specified tasks
- Spoken dialogues recorded
- Cost factors, states, dialog acts automatically logged; ASR accuracy, barge-in hand-labeled
- Users specify task solution via web page
- Users complete User Satisfaction surveys
- Use multiple linear regression to model User Satisfaction as a function of Task Success and Costs; test for significant predictive factors
User Satisfaction: Sum of Many Measures

- Was the system easy to understand? (TTS Performance)
- Did the system understand what you said? (ASR Performance)
- Was it easy to find the message/plane/train you wanted? (Task Ease)
- Was the pace of interaction with the system appropriate? (Interaction Pace)
- Did you know what you could say at each point of the dialog? (User Expertise)
- How often was the system sluggish and slow to reply to you? (System Response)
- Did the system work the way you expected it to in this conversation? (Expected Behavior)
- Do you think you'd use the system regularly in the future? (Future Use)
Performance Functions from Three Systems

- ELVIS User Sat. = 0.21 * COMP + 0.47 * MRS - 0.15 * ET
- TOOT User Sat. = 0.35 * COMP + 0.45 * MRS - 0.14 * ET
- ANNIE User Sat. = 0.33 * COMP + 0.25 * MRS - 0.33 * Help

- COMP: User perception of task completion (task success)
- MRS: Mean (concept) recognition accuracy (cost)
- ET: Elapsed time (cost)
- Help: Help requests (cost)
Evaluation Summary

- Best predictors of User Satisfaction:
  - Perceived task completion
  - mean recognition score (concept accuracy)
- Performance model useful for system development
  - Making predictions about system modifications
  - Distinguishing ‘good’ dialogues from ‘bad’ dialogues
  - As part of a learning model
Now that we have a success metric

- Could we use it to help drive learning?
- Learn an optimal policy or strategy for how the conversational agent should behave
New Idea: Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - Set of states $S$ the agent can be in
  - 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.
What do we mean by actions $A$ and policies $\pi$?

- Kinds of decisions a conversational agent needs to make:
  - When should I ground/confirm/reject/ask for clarification on what the user just said?
  - When should I ask a directive prompt, when an open prompt?
  - When should I use user, system, or mixed initiative?
A threshold is already a policy – a human-designed one!

- Could we learn what the right action is
  - Rejection
  - Explicit confirmation
  - Implicit confirmation
  - No confirmation

- By learning a policy which,
  - given various information about the current state,
  - dynamically chooses the action which maximizes dialogue success
Another strategy decision

- Open versus directive prompts
- When to do mixed initiative

- How we do this optimization?
- Markov Decision Processes
Review: Open vs. Directive Prompts

• **Open prompt**
  • System gives user very few constraints
  • User can respond how they please:
  • “How may I help you?” “How may I direct your call?”

• **Directive prompt**
  • Explicit instructs user how to respond
  • “Say yes if you accept the call; otherwise, say no”
Review: Restrictive vs. Non-restrictive grammars

- Restrictive grammar
  - Language model which strongly constrains the ASR system, based on dialogue state
- Non-restrictive grammar
  - Open language model which is not restricted to a particular dialogue state
Kinds of Initiative

- How do I decide which of these initiatives to use at each point in the dialogue?

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Open Prompt</th>
<th>Directive Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrictive</td>
<td><em>Doesn’t make sense</em></td>
<td><em>System Initiative</em></td>
</tr>
<tr>
<td>Non-restrictive</td>
<td><em>User Initiative</em></td>
<td><em>Mixed Initiative</em></td>
</tr>
</tbody>
</table>
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
Maximum Expected Utility

- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent’s expected utility

- Action A has possible outcome states $Result_i(A)$

- E: agent’s evidence about current state of world

- Before doing A, agent estimates prob of each outcome
  - $P( Result_i(A) | Do(A), E)$

- Thus can compute expected utility:

$$EU(A | E) = \sum_{i} P(Result_i(A) | Do(A), E) U(Result_i(A))$$
Utility (Russell and Norvig)
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

- Levin et al. (2000)
- 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
  - User’s 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

**ad**: a question asking for the day

**am**: a question asking for the month

**adm**: a question asking for the day + month

**af**: 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

  \[ N_i = \text{number of interactions (duration of dialogue)} \]
  \[ N_e = \text{number of errors in the obtained values (0-2)} \]
  \[ N_f = \text{expected distance from goal} \]
  - (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.
- $d=-1$ $m=-1$

$c_1 = -3w_i + 2p_diw_e$

Policy 2 (open)

- $d=0$ $m=0$: What date?
- $d=D$ $m=M$: Goodbye.
- $d=-1$ $m=-1$

$c_2 = -2w_i + 2p_ow_e$

$P_d$ = 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)

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

Policy 2 (open)

\[ c_2 = -2w_i + 2p_o w_e \]
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:
• We can think of a dialogue as a trajectory in state space

\[ S_1 \rightarrow a_1, r_1 \rightarrow S_2 \rightarrow a_2, r_2 \rightarrow S_3 \rightarrow a_3, r_3 \cdots \]

• The best policy \( \pi^* \) is the one with the greatest expected reward over all trajectories

• How to compute a reward for a state sequence?
Reward for a state sequence

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

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

- 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

\[ P(s_{t+1} \mid s_t,s_{t-1},...,s_0,a_t,a_{t-1},...,a_0) = P_T(s_{t+1} \mid s_t,a_t) \]
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 MDP systems were built
- Current direction:
  - 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
Summary

- Dialogue Manager Design
  - Finite State
  - Frame-based
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Utility-based conversational agents
  - MDP, POMDP
- Evaluation