CS 224S / LINGUIST 236
Speech Recognition and Synthesis

Dan Jurafsky

Lecture 18: Advanced Issues in Dialogue Systems: Evaluation (PARADISE) and Markov Decision Processes

IP Notice: Some slides adapted from Julia Hirschberg

Dialogue System Evaluation

- Returning to an issue we just started to touch on a few weeks ago.
- Whenever we design a new algorithm or build a new application, need to evaluate it
- How to evaluate a dialogue system?
- What constitutes success or failure for a dialogue system?

Dialogue System Evaluation

- It turns out we'll need an evaluation metric for two reasons
  - 1) the normal reason: we need a metric to help us compare different implementations
    - can't improve it if we don't know where it fails
    - Can't decide between two algorithms without a goodness metric
  - 2) a new reason: we will need a metric for "how good a dialogue went" as an input to reinforcement learning:
    - automatically improve our conversational agent performance via learning

Evaluating Dialogue Systems

- PARADISE framework (Walker et al '00)
- "Performance" of a dialogue system is affected both by *what* gets accomplished by the user and the dialogue agent and *how* it gets accomplished

Maximize Task Success

Minimize Costs

Efficiency Measures

Qualitative Measures

Slide from Julia Hirschberg

PARADISE evaluation again:

- Maximize Task Success
- Minimize Costs
  - Efficiency Measures
  - Quality Measures
- PARADISE (PARAdigm for Dialogue System Evaluation)

Task Success

- % of subtasks completed
- Correctness of each questions/answer/error msg
- Correctness of total solution
  - Attribute-Value matrix (AVM)
  - Kappa coefficient
- Users' perception of whether task was completed

Task Success

•Task goals seen as Attribute-Value Matrix

ELVIS e-mail retrieval task ***(Walker et al '97)***

*"Find the **time** and **place** of your **meeting** with **Kim**."*

Attribute	*Value*
Selection Criterion	*Kim or Meeting*
Time	*10:30 a.m.*
Place	*2D516*

•Task success can be defined by match between AVM values at end of task with "true" values for AVM

Slide from Julia Hirschberg

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

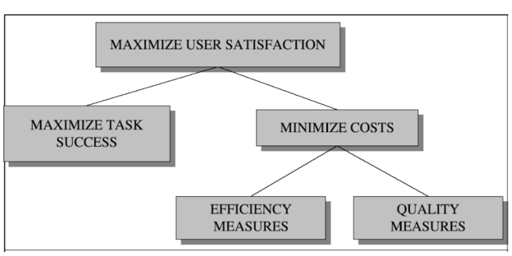
Quality Cost

- # 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

Another key quality cost

- "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"

PARADISE: Regress against user satisfaction



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

Slide from Julia Hirschberg

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

Adapted from Julia Hirschberg

Performance Functions from Three Systems

- ELVIS User Sat.= .21* COMP + .47 * MRS - .15 * ET
- TOOT User Sat.= .35* COMP + .45* MRS - .14*ET
- ANNIE User Sat.= .33*COMP + .25* MRS +.33* Help
  - COMP: User perception of task completion (task success)
  - MRS: Mean (concept) recognition accuracy (cost)
  - ET: Elapsed time (cost)
  - Help: Help requests (cost)

Slide from Julia Hirschberg

Performance Model

- Perceived task completion and mean recognition score (concept accuracy) are consistently significant predictors of User Satisfaction
- 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?
- We'll try to use this metric to help us learn an optimal policy or strategy for how the conversational agent should behave

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 π for what action to take in any particular state.

What do we mean by actions A and policies π?

- 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 few quick slides reviewing Grounding and Confirmation

- Dialogue is a collective act performed by speaker and hearer
- Common ground: set of things mutually believed by both speaker and hearer
- Need to achieve common ground, so hearer must ground or acknowledge speakers utterance.
- Clark (1996):
  - *Principle of closure.* Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it
- Recall: Norman (1988) work, things like elevator buttons that light up
- Need to know whether an action succeeded *or failed*

Clark and Schaefer: Grounding

- Continued attention: B continues attending to A
- Relevant next contribution: B starts in on next relevant contribution
- Acknowledgement: B nods or says continuer like *uh-huh*, *yeah*, assessment (*great!*)
- Demonstration: B demonstrates understanding A by paraphrasing or reformulating A's contribution, or by collaboratively completing A's utterance
- Display: B displays verbatim all or part of A's presentation

Why grounding in conversational agents

- Errors: Speech is a pretty errorful channel
  - Even for humans; so they use grounding to confirm that they heard correctly
- ASR is way worse than humans!
- So dialogue systems need to do even more grounding and confirmation than humans

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's 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

Rejection

- e.g., VoiceXML "nomatch"
- "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 threshhold, reject
  - Above threshold, explicit confirmation
  - If even higher, implicit confirmation
  - Even higher, no confirmation

A threshold is a human-designed policy!

- 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

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 gramamrs

- 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?

Grammar	Open Prompt	Directive Prompt
Restrictive	*Doesn't make sense*	System Initiative
Non-restrictive	User Initiative	Mixed Initiative

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 π 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

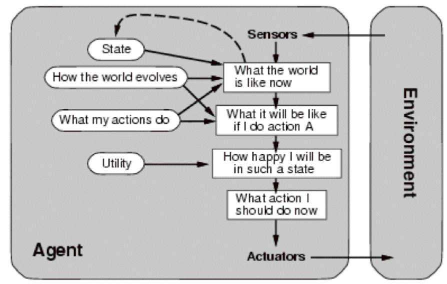
- A utility function
  - maps a state or state sequence
  - onto a real number
  - describing the goodness of that state
  - I.e. the resulting "happiness" of the agent
- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent's expected utility

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:

$$EU(A \mid E) = \sum_i P(Result_i(A) \mid Do(A), E) U(Result_i(A))$$

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
  - 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

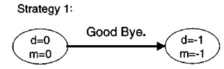
A simple reward function

- 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:
- $C = w_i \times N_i + w_e \times N_e + w_f \times N_f$

3 possible policies

Dumb

Strategy 1:



$C_1 = 1 \cdot W_I + 2 \cdot W_F$

Open prompt

Strategy 2:

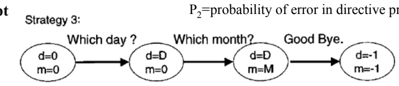
P_1=probability of error in open prompt

$C_2 = 2 \cdot W_I + 2 \cdot P_1 \cdot W_E$

Directive prompt

Strategy 3:

P_2=probability of error in directive prompt



$C_3 = 3 \cdot W_I + 2 \cdot P_2 \cdot W_E$

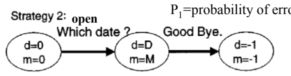
3 possible policies

Strategy 3 is better than strategy 2 when improved error rate justifies longer interaction:

$$p_1 - p_2 > \frac{w_i}{2w_e}$$

Strategy 2: open

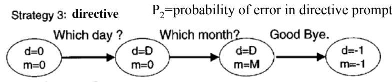
P_1=probability of error in open prompt



$C_2 = 2 \cdot W_I + 2 \cdot P_1 \cdot W_E$

Strategy 3: directive

P_2=probability of error in directive prompt



$C_3 = 3 \cdot W_I + 2 \cdot P_2 \cdot W_E$

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 π^* is harder
- We need reinforcement leraning
- Back to MDPs:

MDP

- We can think of a dialogue as a trajectory in state space

$$s_1 \xrightarrow{}_{a1,r1} s_2 \xrightarrow{}_{a2,r2} s_3 \xrightarrow{}_{a3,r3} \cdots$$

- The best policy π^* 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 γ 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:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

- 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(s1,a1,s2,a2,…,sn)
- 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(s1,a1,s2,a2,…,sn)?
- 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

- Singh, S., D. Litman, M. Kearns, and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of AI Research.
- 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

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