Speech and Language Processing

Chapter 24 of SLP (part 3)
Dialogue and Conversational Agents

Administration

- Homework 3
- Project
  - Test data emailed
  - Presentations should be 10 minutes to allow question time. I will cut them off.
  - Papers should be **NO LONGER THAN** 4 pages (excluding references) using posted templates
- Final Exam
  - All material since scope of last exam
  - Similar format to midterm
    - True/false
    - Short answer (definitions, compare/contrast)
    - Problem solving
- OMET survey

4/24/2017
DA interpretation as statistical classification

- Lots of clues in each sentence that can tell us which DA it is:
  - 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

Statistical classifier model of dialogue act interpretation

- Our goal is to decide for each sentence what dialogue act it is
- This is a classification task (we are making a 1-of-N classification decision for each sentence)
- With \( N \) classes (= number of dialog acts).
- Three probabilistic models corresponding to the 3 kinds of cues from the input sentence:
  - Conversational Structure: Probability of one dialogue act following another \( P(\text{Answer}|\text{Question}) \)
  - Words and Syntax: Probability of a sequence of words given a dialogue act: \( P(\text{“do you”}|\text{Question}) \)
  - Prosody: Probability of prosodic features given a dialogue act: \( P(\text{“rise at end of sentence”}|\text{Question}) \)
An example of dialogue act detection: Correction Detection

- Despite all these clever confirmation/rejection strategies, dialogue systems still make mistakes (Surprise!)
- 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, Litman, & Hirschberg (2000): corrections misrecognized twice as often (in terms of WER) as non-corrections!!!
  - Why?
    - Prosody seems to be largest factor: hyperarticulation
      - “NO, I am DE-PAR-TING from Jacksonville”
A Labeled Toot dialogue  
(Swerts, Litman and Hirschberg)

<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>Um tomorrow</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>Tomorrow</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>Saturday</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

- Build classifiers using features like
  - Lexical information (words “no”, “correction”, “I don’t”, swear words)
  - Prosodics features (various increases in F0 range, pause duration, and word duration that correlation with hyperarticulation)
  - Length
  - ASR confidence
  - LM probability
  - Various dialogue features (repetition)
Generating Dialogue Acts

- Confirmation
- Rejection

Confirmation

- Another reason for grounding
- 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

- 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 threshold, reject
  - Above threshold, explicit confirmation
  - If even higher, implicit confirmation
  - Even higher, no confirmation

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
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
- Users’ perception of whether task was completed
- Learning gains (in tutoring)
Efficiency Cost

- Total elapsed time in seconds or turns
- Number of queries
- Turn correction ratio: 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
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. = .21* COMP + .47 * MRS - .15 * ET  
- TOOT User Sat. = .35* COMP + .45* MRS - .14*ET  
- ANNI E 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)
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?
- In recent work we use this metric to help us 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
    - 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.

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 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

- How we do this optimization?
- Markov Decision Processes
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 till 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:

\[
EU(A \mid E) = \sum_{i} P(\text{Result}_i(A) \mid \text{Do}(A), E)U(\text{Result}_i(A))
\]
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

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

\[ P_o - P_d > \frac{W_i}{2w_e} \]

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?

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

Reward for a state sequence

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

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

- 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

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

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
    - \( p(s'|s,a) = \frac{C(s,s',a)}{C(s,a)} \)
  - 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 $p(s' \mid s, a)$ 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

State of the art

- Hot topics (shared tasks: dialogue state tracking, Ubuntu corpus answer prediction)
  - 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