Speech and Language Processing

Chapter 24: part 2
Dialogue and Conversational Agents

Outline

- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation

- Evaluation
Information-State and Dialogue Acts

- If we want a dialogue system to be more than just form-filling
- Needs to:
  - Decide when the user has asked a question, made a proposal, rejected a suggestion
  - Ground a user's utterance, ask clarification questions, suggestion plans
- Suggests:
  - Conversational agent needs sophisticated models of interpretation and generation
    - In terms of speech acts and grounding
    - Needs 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
Dialogue 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)

Verbmobil Dialogue Acts

<table>
<thead>
<tr>
<th>Action</th>
<th>Transcript</th>
</tr>
</thead>
<tbody>
<tr>
<td>THANK</td>
<td>thanks</td>
</tr>
<tr>
<td>GREET</td>
<td>Hello Dan</td>
</tr>
<tr>
<td>INTRODUCE</td>
<td>It’s me again</td>
</tr>
<tr>
<td>BYE</td>
<td>Alright, bye</td>
</tr>
<tr>
<td>REQUEST-COMMENT</td>
<td>How does that look?</td>
</tr>
<tr>
<td>SUGGEST</td>
<td>June 13th through 17th</td>
</tr>
<tr>
<td>REJECT</td>
<td>No, Friday I’m booked all day</td>
</tr>
<tr>
<td>ACCEPT</td>
<td>Saturday sounds fine</td>
</tr>
<tr>
<td>REQUEST-SUGGEST</td>
<td>What is a good day of the week for you?</td>
</tr>
<tr>
<td>INIT</td>
<td>I wanted to make an appointment with you</td>
</tr>
<tr>
<td>GIVE_REASON</td>
<td>Because I have meetings all afternoon</td>
</tr>
<tr>
<td>FEEDBACK</td>
<td>Okay</td>
</tr>
<tr>
<td>DELIBERATE</td>
<td>Let me check my calendar here</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>Okay, that would be wonderful</td>
</tr>
<tr>
<td>CLARIFY</td>
<td>Okay, do you mean Tuesday the 23rd?</td>
</tr>
</tbody>
</table>
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
- Recognizing illocutionary force will be crucial to building a dialogue agent
- Perhaps we can just look at the form of the utterance to decide?

Can we just use the surface syntactic form?

- YES-NO-Q’s have auxiliary-before-subject syntax:
  - Will breakfast be served on USAir 1557?
- STATEMENTs have declarative syntax:
  - I don’t care about lunch
- COMMAND’s have imperative syntax:
  - Show me flights from Milwaukee to Orlando on Thursday night
Surface form != speech act type

<table>
<thead>
<tr>
<th>Locutionary Force</th>
<th>Illocutionary Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can I have the rest of your sandwich?</td>
<td>Question</td>
</tr>
<tr>
<td>I want the rest of your sandwich</td>
<td>Declarative</td>
</tr>
<tr>
<td>Give me your sandwich!</td>
<td>Imperative</td>
</tr>
</tbody>
</table>

Dialogue act disambiguation is hard! Who’s on First?

Abbott: Well, let's see, we have on the bags, Who's on first, What's on second, I Don't Know is on third.  
   Intended:  
   Understood:  

Costello: Well, then, who's playing first?.  
   Intended:  
   Understood:
Dialogue act ambiguity

- Who’s on first?
  - STATEMENT (intended)
  - Or
  - INFO-REQUEST (understood)

- Who’s playing first?
  - INFO-REQUEST (intended)
  - or
  - CHECK (understood)

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 beginning 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

- 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(Answer|Question)
  - Words and Syntax: Probability of a sequence of words given a dialogue act: P("do you" | Question)
  - Prosody: probability of prosodic features given a dialogue act: P("rise at end of sentence" | 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 On which day of the week do you want to leave?</td>
<td></td>
</tr>
<tr>
<td>S Sorry, I didn’t hear anything from you. Please tell me your departure date</td>
<td></td>
</tr>
<tr>
<td>U <em>Um tomorrow</em></td>
<td>NA</td>
</tr>
<tr>
<td>S You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'...</td>
<td>REP</td>
</tr>
<tr>
<td>U <em>Tomorrow</em></td>
<td></td>
</tr>
<tr>
<td>S You can say the departure date. For example, you can say 'today' or 'tomorrow'. Or you can say 'Monday', 'Tuesday', 'Wednesday'...</td>
<td>PAR</td>
</tr>
<tr>
<td>U <em>Saturday</em></td>
<td></td>
</tr>
<tr>
<td>S Do you want to leave on Saturday?</td>
<td></td>
</tr>
<tr>
<td>U Yes</td>
<td></td>
</tr>
<tr>
<td>S What time do you want to leave?</td>
<td></td>
</tr>
<tr>
<td>U Eight o’clock A M</td>
<td></td>
</tr>
<tr>
<td>S Do you want to leave around ten o’clock p-m</td>
<td>OMIT</td>
</tr>
<tr>
<td>U Eight o’clock</td>
<td></td>
</tr>
<tr>
<td>S Do you want to leave around ten o’clock p-m</td>
<td>OMIT</td>
</tr>
<tr>
<td>U No</td>
<td></td>
</tr>
<tr>
<td>S What time do you want to leave?</td>
<td></td>
</tr>
<tr>
<td>U Eight o’clock A M</td>
<td>ADD</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

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

- 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

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

- 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

- Key point about SLP.
- Whenever we design a new algorithm or build a new application, need to evaluate it
- Two kinds of evaluation
  - Extrinsic: embedded in some external task
  - Intrinsic: some sort of more local evaluation.

- 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

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”
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)

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