Spoken Dialogue Systems: Managing Interaction

Julia Hirschberg

CS 4706
Outline

• ‘Rules’ of Human-Human Conversation
  – Turn-taking
  – Speech Acts
  – Grounding
• Dialogue Management in SDS
  – Types of Dialogue Management
  – Varieties of Initiative
• VoiceXML
Turn-taking

• Dialogue is characterized by turn-taking.
  – A:
  – B:
  – A:
  – B:
  – ...

• Resource allocation problem

• How do speakers know when to take the floor?
  – Total amount of overlap relatively small (5% - Levinson 1983)
  – But there is very little pause
  – Must be a way to know who should talk and when
Turn-taking rules

• At each transition-relevance place (TRP) of each turn:
  – a) If during this turn the current speaker has selected B as the next speaker, then B must speak next.
  – b) If the current speaker does not select the next speaker, any other speaker may take the next turn.
  – c) If no one else takes the next turn, the current speaker may take the next turn.
Implications of Subrule a

- For some utterances, current speaker selects next speaker
  - Adjacency pairs
    - Question/answer
    - Greeting/greeting
    - Compliment/downplayer
    - Request/grant
- Silence between 2 parts of adjacency pair is different than silence after
  - A: Is there something bothering you or not?
  - (1.0)
  - A: Yes or no?
  - (1.5)
  - A: Eh?
  - B: No.
Speech Acts

• Austin (1962): An utterance is a kind of action
• Clear case: performatives
  – I name this ship the Titanic
  – I second that motion
  – I bet you five dollars it will snow tomorrow
• Performative verbs (name, second, bet…)
• Austin’s idea: not just these verbs
Each utterance is 3 acts

• **Locutionary act**: the utterance of a sentence with a particular meaning

• **Illocutionary act**: the act of asking, answering, promising, etc., in uttering a sentence.

• **Perlocutionary act**: the (often intentional) production of certain effects upon the thoughts, feelings, or actions of addressee in uttering a sentence.
Locutionary vs. Illocutionary vs. Perlocutionary

• “You can’t do that!”
• Illocutionary force:
  – Protest
• Perlocutionary force:
  – Intent to annoy addressee
  – Intent to stop addressee from doing something
Illocutionary Acts

• How many are there?
• What are they?
• How do we decide?
Some Ideas from Searle (1975): Speech Acts

- **Assertives:** Commitments by the speaker to something’s being the case
  - suggesting, putting forward, swearing, boasting, concluding
- **Directives:** Attempts by the speaker to get the addressee to do something
  - asking, ordering, requesting, inviting, advising, begging
- **Commissives:** Commitments by the speaker to some future course of action
  - promising, planning, vowing, betting, opposing
- **Expressives:** Expressions of the psychological state of the speaker about a state of affairs
  - thanking, apologizing, welcoming, deploring
- **Declarations:** Utterances by the speaker that themselves bring about a different state of the world
  - I resign; You’re fired; I now pronounce you…}
Grounding

- Assumption: Dialogue is a collective act performed by speaker (S) and hearer (H)
- **Common ground**: set of things mutually believed by both speaker and hearer
- S and H need to achieve common ground to achieve successful communication, so H must ground or acknowledge S’s utterance
- Clark (1996):
  - **Principle of closure**. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it
  - True in HCI as well (Norman, 1988)
  - Need to know whether an action succeeded or failed
Clark and Schaefer: Types of 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
A human-human conversation

C1: ... I need to travel in May.
A1: And, what day in May did you want to travel?
C2: OK uh I need to be there for a meeting that’s from the 12th to the 15th.
A2: And you’re flying into what city?
C3: Seattle.
A3: And what time would you like to leave Pittsburgh?
C4: Uh hmm I don’t think there’s many options for non-stop.
A4: Right. There’s three non-stops today.
C5: What are they?
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.
C6: OK I’ll take the 5ish flight on the night before on the 11th.
C7: OK.
Grounding examples

• Display:
  – C: I need to travel in May
  – A: And, what day in May did you want to travel?

• Acknowledgement
  – C: He wants to fly from Boston
  – A: mm-hmm
  – C: to Baltimore Washington International
  – [Mm-hmm (usually transcribed “uh-huh”) is a backchannel, continuer, or acknowledgement token]
• Acknowledgement + next relevant contribution
  – And, what day in May did you want to travel?
  – And you’re flying into what city?
  – And what time would you like to leave?

• The *and* indicates to the client that agent has successfully understood answer to the last question.
Grounding negative responses
From Cohen et al. (2004)

• System: Did you want to review some more of your personal profile?
• Caller: No.
• System: Okay, what’s next?

Good!

• System: Did you want to review some more of your personal profile?
• Caller: No.
• System: What’s next?

Bad!
Grounding and Dialogue Systems

- Grounding is not just a useful fact about humans
- Key to designing a good conversational agent
- Why?
Grounding and Dialogue Systems

• Grounding is not just a tidbit about humans
• Is key to design of conversational agent
• Why?
  – HCI researchers find users of speech-based interfaces are confused when system doesn’t give them an explicit acknowledgement signal
  – Stifelman et al. (1993), Yankelovich et al. (1995)
Dialogue Manager

• Controls the architecture and structure of dialogue
  – Takes input from ASR/NLU components
  – Maintains some sort of state
  – Interfaces with Task Manager
  – Passes output to NLG/TTS modules
Architectures for Dialogue Management

• Finite State
• Frame-based
• Information State
  – Markov Decision Processes
• AI Planning
Finite-State Dialogue Management

• A trivial airline travel system
  – Ask the user for a departure city
  – For a destination city
  – For a time
  – Whether the trip is round-trip or not
Finite State Dialogue Manager

What city are you leaving from?

Where are you going?

What date do you want to leave?

Is it a one-way trip?

Yes

Do you want to go from <FROM> to <TO> on <DATE>?

No

What date do you want to return?

Yes

Do you want to go from <FROM> to <TO> on <DATE> returning on <RETURN>?

No

Book the flight
Finite-state Dialogue Managers

• System completely controls the conversation with the user
• Asks the user a series of questions
• Ignores (or misinterprets) anything the user says that is *not* a direct answer to the system’s questions
Dialogue Initiative

• Systems that control conversation like this are system initiative or single initiative
• "Initiative": who has control of conversation
• In normal human-human dialogue, initiative shifts back and forth between participants
System Initiative SDS

• Advantages:
  – Simple to build
  – User always knows what they can say next
  – System always knows what user can say next
    • Known words: Better performance from ASR
    • Known topic: Better performance from NLU
  – Ok for very simple tasks (entering a credit card, or login name and password)

• Disadvantage:
  – Too limited
Major Problems with System Initiative

• Real dialogue involves give and take
• In travel planning, e.g., users might want to say something that is not the direct answer to the question
• E.g.
  – System: What city do you want to leave from?
  – User1: Hi, I’d like to fly from Seattle Tuesday morning
  – User2: I want a flight from Milwaukee to Orlando one way leaving after 5 p.m. on Wednesday.
One Option: Single initiative + Universals

• Give users a little more flexibility by adding universal commands
• **Universals**: commands you can say anywhere
• Augment every state of FSA with these options:
  – Help
  – Start over
  – Correct
• This describes many implemented systems
• But still doesn’t allow user to say what they want to say
User Initiative

• User directs the system
• Generally, user asks a single question, system answers
• System can’t ask questions back, engage in clarification dialogue, confirmation dialogue
• Used for simple database queries
  – User asks a question, system gives an answer
  – E.g., Web search is user initiative dialogue
Mixed Initiative

• Conversational initiative can shift between system and user
• Simplest kind of mixed initiative: use structure of a frame to guide dialogue: goal is fill in the slots by asking the questions
  – Slot Question
  – ORIGIN What city are you leaving from?
  – DEST Where are you going?
  – DEPT DATE What day would you like to leave?
  – DEPT TIME What time would you like to leave?
  – AIRLINE What is your preferred airline?
Defining Mixed Initiative

• Mixed Initiative could mean
  – User can arbitrarily take or give up initiative in various ways
    • Only possible in very complex plan-based dialogue systems
    • No commercial implementations
    • Important research area
  – Something simpler and quite specific
Mixed-Initiative Frame-based Systems

• User can answer multiple questions at once
• System asks questions to fill in remaining slots
• When frame is filled, we’re done!
  – Do database query
• If user answers 3 questions at once, system fills in those slots and doesn’t ask the slot questions
• Advantages:
  – Avoid strict constraints on order of the finite-state architecture
  – Faster but riskier!
Systems with Multiple frames

• E.g., flights, hotels, rental cars
• Subframes, e.g. Flight legs: Each flight can have multiple legs, which might need to be discussed separately
• Multiple instantiations: e.g. Presenting multiple flights meeting users constraints
  – Slots like 1ST_FLIGHT or 2ND_FLIGHT so user can ask “how much is the second one”
• General route information:
  – Which airlines fly from Boston to San Francisco?
• Airfare practices:
  – Do I have to stay over Saturday to get a decent airfare?
Problems with Multiple Frames

• Need to be able to switch from frame to frame – how?
  – Based on what user says?
  – Based on likelihood of frame sequence

• Disambiguate which slot of which frame an input is supposed to fill, then switch dialogue control to that frame.

• Main implementation: production rules
  – Different types of inputs cause different productions to fire
  – Each of which can flexibly fill in different frames
  – Can also switch control to different frame
True Mixed Initiative

C₁: ... I need to travel in May.
A₁: And, what day in May did you want to travel?
C₂: OK uh I need to be there for a meeting that’s from the 12th to the 15th.
A₂: And you’re flying into what city?
C₃: Seattle.
A₃: And what time would you like to leave Pittsburgh?
C₄: Uh hhm I don’t think there’s many options for non-stop.
A₄: Right. There’s three non-stops today.
C₅: What are they?
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.
C₆: OK I’ll take the 5ish flight on the night before on the 11th.
A₆: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air
    flight 115.
C₇: OK.
Implementing a Mixed Initiative System

- Two criteria:
  - Open prompts vs. directive prompts
  - Restrictive versus non-restrictive grammar
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”
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
# Definition of Mixed Initiative

<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>
VoiceXML

- Voice eXtensible Markup Language
- An XML-based dialogue design language
- Makes use of ASR and TTS
- Deals well with simple, frame-based mixed initiative dialogue.
- Most common in commercial world (too limited for research systems)
- But useful to get a handle on the concepts
Voice XML

• Each dialogue is a `<form>`.
  (Form is the VoiceXML word for frame)
• Each `<form>` generally consists of a sequence of `<field>`s, with other commands
Sample VXML Form

<form>
  <field name="transporttype">
    <prompt>
      Please choose airline, hotel, or rental car.  </prompt>
    <grammar type="application/x=nuance-gsl">
      [airline hotel "rental car"]
    </grammar>
  </field>
  <block>
    <prompt>
      You have chosen <value expr="transporttype">. </prompt>
  </block>
</form>
VoiceXML interpreter

• Walks through a VXML form in document order
• Iteratively selecting each item
• If multiple fields, visit each one in order
• Special commands for events
Reprompting Forms

<noinput>
I'm sorry, I didn't hear you.  <reprompt/>
</noinput>
- “noinput” means silence exceeds a timeout threshold

<nomatch>
I'm sorry, I didn't understand that.  <reprompt/>
</nomatch>

- “nomatch” means confidence value for utterance is too low
- notice “reprompt” command
Welcome Form

<form>
    <block> Welcome to the air travel consultant. </block>
    <field name="origin">
        <prompt> Which city do you want to leave from? </prompt>
        <grammar type="application/x=nuance-gsl">
            [(san francisco) denver (new york) barcelona]
        </grammar>
        <filled>
            <prompt> OK, from <value expr="origin"> </prompt>
        </filled>
    </field>
</form>

- “filled” tag is executed by interpreter as soon as field filled by user
<field name="destination">
  <prompt> And which city do you want to go to? </prompt>
  <grammar type="application/x=nuance-gsl">(san francisco) denver (new york) barcelona</grammar>
</field>

<field name="departdate" type="date">
  <prompt> And what date do you want to leave? </prompt>
</field>
Summing Up

<block>
<prompt> OK, I have you are departing from
  <value expr="origin"> to <value expr="destination"> on <value expr="departdate">
  </prompt>
  send the info to book a flight...
</block>
</form>
Summary

• Human-human conversation
  – Turn-taking
  – Speech Acts
  – Grounding
  – Error Handling and Help

• Dialogue Manager Design
  – Finite State
  – Frame-based
  – Initiative: User, System, Mixed

• VoiceXML
Next Class

• Information State and Dialogue Acts