Speech and Language Processing

Chapter 24: part 2
Dialogue and Conversational Agents

Outline

- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State vs Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Information-State
  - Dialogue-Act Detection
- Evaluation (next time)
**Example SDS Architecture**

- **Speech recognition**: I am looking for a coffee shop near Pitt
  - **Natural language understanding**: Type = Coffee, Area = University of Pittsburgh
  - **Dialogue manager**: Offer(name=CrazyMochaOakland)
  - **Natural language generation**: Crazy Mocha is near the university
  - **Text-to-speech or recording**: Offer(name=CrazyMochaOakland)

---

**Speech recognition**

- Or ASR (Automatic Speech Recognition)
  - Speech to words
  - Input: acoustic waveform
  - Output: string of words
  - Basic components:
    - a **recognizer for phones**, small sound units like [k] or [ae].
    - a **pronunciation dictionary** like cat = [k ae t]
    - a **grammar** or **language model** telling us what words are likely to follow what words
    - A **search algorithm** to find the best string of words

4/20/2017
Natural Language Understanding

- Or “NLU”
- Or “Computational semantics”
- There are many ways to represent the meaning of sentences
- For speech dialogue systems, most common is “Frame and slot semantics”.

An example of a frame

- Show me morning flights from Boston to SF on Tuesday.
  SHOW:
  FLIGHTS:
    ORIGIN:
      CITY: Boston
      DATE: Tuesday
      DEPART-TIME:
    DEST:
      CITY: San Francisco
How to generate this semantics?

- Many methods,
- Simplest: “semantic grammars”
- CFG in which the LHS of rules is a semantic category:
  - LIST -> show me | I want | can I see|...
  - DEPARTTIME -> (after|around|before) HOUR | morning | afternoon | evening
  - HOUR -> one|two|three...|twelve (am|pm)
  - FLIGHTS -> (a) flight|flights
  - ORIGIN -> from CITY
  - DESTINATION -> to CITY
  - CITY -> Boston | San Francisco | Denver | Washington

Semantics for a sentence

**LIST**    **FLIGHTS**    **ORIGIN**
Show me    flights    from Boston

**DESTINATION**    **DEPARTDATE**
to San Francisco on Tuesday

**DEPARTTIME**
morning
**Generation and TTS**

- **Generation component**
  - Chooses concepts to express to user
  - Plans out how to express these concepts in words
  - Assigns any necessary prosody to the words
- **TTS component**
  - Takes words and prosodic annotations
  - Synthesizes a waveform

**Generation Component**

- **Content Planner**
  - Decides what content to express to user
    - (ask a question, present an answer, etc)
  - Often merged with dialogue manager
- **Language Generation**
  - Chooses syntactic structures and words to express meaning.
  - Simplest method
    - All words in sentence are prespecified!
    - “Template-based generation”
    - Can have variables:
      - What time do you want to leave CITY-ORIG?
      - Will you return to CITY-ORIG from CITY-DEST?
More sophisticated language generation component

- Natural Language Generation
- This is a field, like Parsing, or Natural Language Understanding, or Speech Synthesis, with its own (small) conference
- Approach:
  - Dialogue manager builds representation of meaning of utterance to be expressed
  - Passes this to a “generator”
  - Generators have three components
    - Sentence planner
    - Surface realizer
    - Prosody assigner

Architecture of a generator for a dialogue system
(after Walker and Rambow 2002)
Statistical Generation

- More recent approaches
  - Extractive and abstractive

HCI constraints on generation for dialogue: “Coherence”

- Discourse markers and pronouns (“Coherence”):
  1. Please say the date.
  2. Please say the start time.
  3. Please say the duration...
  4. Please say the subject...
  2. First, tell me the date.
  Next, I’ll need the time it starts.
  Thanks. <pause> Now, how long is it supposed to last?
  Last of all, I just need a brief description

Bad!

Good!
HCI constraints on generation for dialogue: coherence (II): tapered prompts

- Prompts which get incrementally shorter:

  - System: Now, what's the first company to add to your watch list?
  - Caller: Cisco
  - System: What's the next company name? (Or, you can say, “Finished”)
  - Caller: IBM
  - System: Tell me the next company name, or say, “Finished.”
  - Caller: Intel
  - System: Next one?
  - Caller: America Online.
  - System: Next?
  - Caller: ...
Four architectures for dialogue management

- Finite State
- Frame-based
- Information State
  - (Partially Observable) Markov Decision Processes
- AI Planning

Finite-State Dialogue Mgmt

- Consider 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 managers

- System completely controls the conversation with the user.
- It asks the user a series of questions
- Ignoring (or misinterpreting) 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**

- Systems which completely control the conversation at all times are called system initiative.
- **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
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 question, system gives answer
- Web search is user initiative dialogue.

Problems with System Initiative

- Real dialogue involves give and take!
- In travel planning, users might want to say something that is not the direct answer to the question.
- For example answering more than one question in a sentence:
  - Hi, I’d like to fly from Seattle Tuesday morning
  - I want a flight from Milwaukee to Orlando one way leaving after 5 p.m. on Wednesday.
Single initiative + universals

- We can give users a little more flexibility by adding universal commands
- Universals: commands you can say anywhere
- As if we augmented every state of FSA with these
  - Help
  - Start over
  - Correct
- This describes many implemented systems
- But still doesn't allow users to say what they want to say

Mixed Initiative

- Conversational initiative can shift between system and user
- Simplest kind of mixed initiative: use the structure of the frame itself to guide dialogue

- **Slot**
  - 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?
Frames are mixed-initiative

- User can answer multiple questions at once.
- System asks questions of user, filling any slots that user specifies
- When frame is filled, do database query
- If user answers 3 questions at once, system has to fill slots and not ask these questions again!
- Anyhow, we avoid the strict constraints on order of the finite-state architecture.

Multiple frames

- flights, hotels, rental cars
- Flight legs: Each flight can have multiple legs, which might need to be discussed separately
- Presenting the flights (If there are multiple flights meeting users constraints)
  - It has slots like 1ST_FLIGHT or 2ND_FLIGHT so user can ask “how much is the second one”
Multiple Frames

- Need to be able to switch from frame to frame
- Based on what user says.
- 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

Defining Mixed Initiative

- Mixed Initiative could mean
  - User can arbitrarily take or give up initiative in various ways
    - This is really only possible in very complex plan-based dialogue systems
    - No commercial implementations
    - Important research area
  - Something simpler and quite specific which we will define in the next few slides
## 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 hum 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. |

## How mixed initiative is usually defined

- First we need to define two other factors
- 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 doc

```xml
<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

---

**Another vxml doc (1)**

```xml
<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
```
Summary: 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.

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

THANK: thanks
GREET: Hello Dan
INTRODUCE: It's me again
BYE: Alright, bye
REQUEST-COMMENT: How does that look?
SUGGEST: June 13th through 17th
REJECT: No, Friday I’m booked all day
ACCEPT: Saturday sounds fine
REQUEST-SUGGEST: What is a good day of the week for you?
INIT: I wanted to make an appointment with you
GIVE_REASON: Because I have meetings all afternoon
FEEDBACK: Okay
DELIBERATE: Let me check my calendar here
CONFIRM: Okay, that would be wonderful
CLARIFY: Okay, do you mean Tuesday the 23rd?

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

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>

4/20/2017
Indirect speech acts

- Utterances which use a surface statement to ask a question
- Utterances which use a surface question to issue a request