Dialogue and Conversational Agents

Regina Barzilay

MIT

December, 2005
A travel dialog: Communicator

S1 Hello. You’ve reached the [Communicator.] Tell me your name
U2 Hi I’d like to fly to Seattle Tuesday morning
S3 Travelling to Seattle on Tuesday, August 11th in the morning.
U4 Yes.
S5 Your full name?
U6 John Doe
Call routing: ATT HMIHY

S1  How may I help you?
U2  can you tell me how much it is to tokyo?
S3  You want to know the cost of a call?
U4  yes that’s right
S5  Please hold on for rate information
A tutorial dialogue: ITSPoke

S1 What force acts on all objects within close proximity to earth?
U2 uh the force of gravity
S3 Fine. Besides the Earth’s gravitational force, are there any other forces?
U4 no
Outline

• Analyzing Human Conversations

• Architecture of Dialogue Systems
  – ASR
  – NLU
  – Generation
  – Dialogue Manager

• Statistical model for the NLU component
Analyzing Human Conversations

- Human data is used to inform design of conversational systems
  - scheduling assistant
  - cross-language information access
  ...

- Computational questions:
  - how to represent structural information in dialogue?
  - how to compute this representation?
Speech Acts

• Austin (1962): An utterance is a kind of action

• Clear case: performatives
  – I name this ship the Titanic
  – I bet you five dollars it will snow tomorrow

• Austin’s idea: not just verbs
One utterance — three 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
Example

You can’t do that!

• Locutionary force:
  – Imperative

• Illocutionary force:
  – Protesting

• Perlocutionary act:
  – Intent to annoy addressee
  – Intent to stop addresses from doing something
Five classes of Speech Acts (Searle, 1975)

- **Assertives:** committing the speaker to something’s being the case (*suggesting, putting forward, swearing, boasting*)

- **Directives:** attempts by the speaker to get the addressee to do something (*asking, ordering, requesting*)

- **Commisives:** committing the speaker to future course of action (*promising, planning, vowing, betting, opposing*)

- **Expressives:** expressing the psychological state of the speaker about a state of affairs (*thanking, apologizing, welcoming, deploring*)

- **Declarations:** bringing about a different state of the world via the utterance (*I resign; You’re fired*)
Dialogue Acts

• An act with associated structural information related to its dialogue function

• Multiple classification schemes have been developed in the past

• These schemes combine ideas from Searle, Austin and others, but details may change from one domain to another

• Verbmobil task
  – Two-party scheduling dialogues
  – Speakers were asked to plan a meeting at some future date
DAMSL: forward looking func.

**STATEMENT**  a claim made by the speaker

**INFO-REQUEST**  a question by the speaker

**CHECK**  a question for confirming information

**INFLUENCE-ON-ADDRESSEE**  (Searle’s directives)

**OPEN-OPTION**  a weak suggestion or listing of options

**ACTION-DIRECTIVES**  on actual command

**INFLUENCE-ON-SPEAKER**  (Austin’s commissives)

**OFFER**  speaker offers to do something

**COMMIT**  speaker is committed to doing something
DAMSL: backward looking func.

**STATEMENT**  speaker’s response to previous proposal
**ACCEPT**  accepting the proposal
**ACCEPT-PART**  accepting some part of the proposal
**MAYBE**  neither accepting nor rejecting the proposal
**REJECT-PART**  rejecting some part of the proposal
**REJECT**  rejecting the proposal
**HOLD**  putting off response
**ANSWER**  answering a question
Example

assert I need to travel in May
infor-req, ack And, what day in May did you want to travel?
assert, answer OK uh I need to be there for a meeting that’s from the 1st
info-req, ack And you are flying into what city?
assert, answer Seattle
THANK          Thanks
GREET          Hello Dan
GREET          It’s me again
INIT           I wanted to make an appointment with you
REQ-COMMENT   How does that look?
SUGGEST        June 13th through 17th
REJECT         No, Friday I’m booked all day
REQ-SUGGEST   What is a good day of the week for you?
GIVE-REASON    Because I have meetings all afternoon
Hello, Mrs. Klein, we should arrange an appointment for the meeting.

Well, I suggest in January, between the 15th and the 19th.

Oh, that is really inconvenient.

... 

Very good, that suits me too, I can make it.
Automatic Interpretation of Dialogue Acts

- Task: automatic identification of 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 1555?

- **STATEMENTS**
  - I don’t care about lunch

- **COMMANDs** have imperative syntax:
  - Show me flights from Boston to NY on Monday night
<table>
<thead>
<tr>
<th>Can I have your coffee?</th>
<th>Question</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>I want your coffee</td>
<td>Declarative</td>
<td>Request</td>
</tr>
<tr>
<td>Give me your coffee</td>
<td>Imperative</td>
<td>Request</td>
</tr>
</tbody>
</table>
• 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
Indirect speech acts

- Utterances which use a surface statement to ask a question
- Utterances which use a surface question to issue a request
Sequence modeling for DA interpretation

- 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
HMM model for DA interpretation

- A dialogue is an HMM
- The hidden states are the dialogue acts
- The observation sequences are sentences
  - Each observation is one sentence
- The observation likelihood model is a word N-gram
HMMs for DA interpretation

- Goal of HMM model:
  
  to compute labeling of dialogue acts
  
  $D = d_1, d_2, \ldots, d_n$ that is most probable given evidence $E$

  $D^* = \arg\max_D P(D|E) = \arg\max_D \frac{P(E|D)P(D)}{P(E)}$

  $= \arg\max_D P(E|D)P(D)$
HMMs for DA interpretation

\[ D^* = \arg\max_D P(E|D)P(D) \]

- Let \( W \) be word sequence in sentence and \( F \) be prosodic feature sequence
- Simplifying independence assumption:
  \[ P(E|D) = P(F|D)P(W|D) \]
- (What are the implications of this?)
  \[ D^* = \arg\max_D P(F|D)P(W|D)P(D) \]
HMMs for DA interpretation

\[ D^* = \arg \max_D P(F|D)P(W|D)P(D) \]

- \( P(D) \): probability of sequence of dialogue acts
- \( P(F|D) \): probability of prosodic sequence given one dialogue act
- \( P(W|D) \): probability of word string in a sentence given dialogue act
Estimating $P(D)$

Markov assumption: each dialogue act depends only on previous $N$ ($N = 3$)

$$P(D) = \prod_{i=2}^{N} P(d_i|d_{i-1}, \ldots, d_{i-N+1})$$
Estimating $P(W|D)$

- Each dialogue act has different words
  - Questions have are you, do you, etc.

$$P(W|D) = \prod_{i=2}^{N} P(w_i|w_{i-1}, \ldots, w_{i-N+1}, d_i)$$
Estimating $P(F|D)$

- A classifier (decision tree) trained on simple acoustically-based prosodic features:
  - Average energy at different places in utterance
  - Various duration measures
- Prosody allows us to distinguish between various DAs:
  - Statement
  - Yes-No-Question
  - Declarative-Question
Estimating $P(F|D)$

- Classifier give posterior $p(D|F)$
- We need $p(F|d)$ to fit into HMM

$$p(d|F) = \frac{p(F|d)p(d)}{p(F)}$$

- Rearranging terms to get a likelihood:

$$\frac{p(F|d)}{p(F)} = \frac{P(d|F)}{p(d)}$$
Final HMM equation for DA interpretation

\[ D^* = \arg \max_D P(F|D) P(W|D) P(D) \]

\[
\prod_{i=2}^{M} P(d_i | d_{i-1}, \ldots, d_{i-M+1}) \prod_{i=2}^{N} \frac{P(d_i|F)}{p(d_i)} \prod_{i=2}^{N} P(w_i | w_{i-1}, \ldots, w_{i-N+1}, d_i)
\]

- We can use Viterbi decoding to find \( D^* \)
- In real dialogue systems, obviously can’t use future dialogue acts, so predict up to current act
- In rescoring passes (for example for labeling human-human dialogues for meeting summarization), can use future info.
Outline

• Analyzing Human Conversations

• Architecture of Dialogue Systems
  – ASR
  – NLU
  – Generation
  – Dialogue Manager

• Statistical model for the NLU component
Automatic Speech Recognition engine

• Based on a standard ASR engine
  – Maps speech to words

• Has specific characteristics for dialogue
  – Language model could depend on where we are in the dialogue
  – Could make use of the fact that we are talking to the same human over time (speaker adaptation)
  – Confidence values (we want to know if the system misunderstood the human)
LM for Dialogue Systems

- Language models for dialogue are often based on hand-written Context-Free or finite-state grammars rather than N-grams

- We can have LM specific to a dialogue state
  - If system just asked “What city are you departing from?”
  - LM should predict:
    * City names only
    * FSA: (I want to (leave|depart)) (from) [CITYNAME]
    * N-grams trained on answers to “Cityname” questions from labeled data
Natural Language Understanding

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

*Show me morning flights from Boston to NY on Tuesday*

SHOW:

FLIGHTS:

ORIGIN

CITY: Boston

DATE: Tuesday

TIME: morning

DEST
How to generate this semantics

• Design a semantic grammar for a domain
  LIST → show me | I want | can I see |
  DEPARTTIME → (after|before|around) HOUR | morning | evening
  HOUR → one | . . . | twelve | (am|pm)
  FLIGHTS → a (flight) — flights
  ORIGIN → from CITY
  DESTINATION → to CITY
  CITY → Boston | San Francisco

• Use a parser to map a sentence into a semantic representation (we will see an example of statistical mapping later in the lecture)
Generation and Text-to-Speech Synthesis

• Generation component
  – Chooses concepts to express to user
  – Plans 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

• Chooses syntactic structures and words to express semantic predicates (provided by dialogue manager)

• Typically implemented using template-based method
  – all concepts are associated with corresponding templates
  – each template has variables instantiated during the generation process

  *What time do you want to leave CITY-ORIG?*

  – LM scores are used to select among alternatives
Dialogue Manager

- Takes input from ASR/NLU components
- Maintains some sort of state
- Interfaces with Task Manager
- Passes output to NLG/TTS modules
Dialogue Manager

- Finite State
  - Single-initiative: system completely controls the conversation with the user
  - Implementation: cascade of FSAs

- Frame-based
  - Mixed-initiatives:
    - system asks questions of user, filling any slots that user specifies
    - when frame is filled, do database query
  - Implementation: production rules that switch control among various frames

- Planning Agents (next time)

- Markov Decision Processes (next time)
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

YES

What date do you want to return?

NO

YES

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

[Book the flight]

NO
Outline

• Analyzing Human Conversations
• Architecture of Dialogue Systems
  – ASR
  – NLU
  – Generation
  – Dialogue Manager
• Statistical model for the NLU component
Statistical NLU component

- A fully statistical approach to natural language interfaces
- Task: map a sentence + context to a database query

User: Show me flights from NY to Boston, leaving tomorrow
System: [returns a list of flights]

<table>
<thead>
<tr>
<th>Show:</th>
<th>(Arrival-time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>(City ”NY“)</td>
</tr>
<tr>
<td>Destination:</td>
<td>(City ”Boston“)</td>
</tr>
<tr>
<td>Date:</td>
<td>(November 27th, 2003)</td>
</tr>
</tbody>
</table>
Representation

- $W =$ input sentence
- $H =$ history (some representation of previous sentences)
- $T =$ a parse tree for $W$
- $F, S =$ a context-independent semantic representation for $W$
- $M =$ a context-dependent representation for $W$ ($M$ depends on both $F$, $S$ and $H$)
Example

\( W = \) input sentence; \( H = \) history; \( T = \) a parse tree for \( W; \ F, S = \) a context independent semantic representation for \( W; \ M = \) a context-dependent semantic representation for \( W \)

User: Show me flights from Newark or New York to Atlanta, leaving tomorrow

System: returns a list of flights

User: When do the flights that leave from Newark arrive in Atlanta

\( W = \) When do the flights that leave from Newark arrive in Atlanta

\[
\begin{array}{|c|c|}
\hline
\text{Show:} & (\text{flights}) \\
\hline
\text{Origin} & (\text{City ”NY“}) \text{ or } (\text{City ”NY“}) \\
\hline
\text{Destination:} & (\text{City ”Atlanta“}) \\
\hline
\text{Date:} & \text{(November 27th, 2003)} \\
\hline
\end{array}
\]
Example

\( W = \) input sentence; \( H = \) history; \( T = \) a parse tree for \( W \); \( F, S = \) a context independent semantic representation for \( W \); \( M = \) a context-dependent semantic representation for \( W \)

User: Show me flights from Newark or New York to Atlanta, leaving tomorrow
System: returns a list of flights
User: When do the flights that leave from Newark arrive in Atlanta

\( W = \) When do the flights that leave from Newark arrive in Atlanta

\[ F, S = \begin{align*}
\text{Show:} & \quad (\text{Arrival-time}) \\
\text{Origin} & \quad (\text{City “Newark”}) \\
\text{Destination:} & \quad (\text{City ”Atlanta”})
\end{align*} \]
<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
</table>
| **H=**
| **Show:** (flights)  
**Origin:** (City "NY") or (City "NY")  
**Destination:** (City "Atlanta")  
**Date:** (November 27th, 2003) |
| **F,S=**
| **Show:** (Arrival-time)  
**Origin:** (City "Newark")  
**Destination:** (City "Atlanta") |
| **M=**
| **Show:** (Arrival-time)  
**Origin:** (City "Newark")  
**Destination:** (City "Atlanta")  
**Date:** (November 27th, 2003) |
Each non-terminal has a syntactic and semantic tag, e.g., city/npr
Building a Probabilistic Model

- Basic goal: build a model of $P(M|W, H)$ – probability of a context-dependent interpretation, given a sentence and a history

- We’ll do by building a model of $P(M, W, F, T, S|H)$, giving

$$P(M, W|H) = \sum_{F,T,S} P(M, W, F, T, S|H)$$

and

$$\arg\max_M P(M|W, H) = \arg\max_M P(M, W|H)$$

$$= \arg\max_M \sum_{F,T,S} P(M, W, F, T, S|H)$$
Building a Probabilistic Model

Our aim is to estimate $P(M, W, F, T, S|H)$

- Apply Chain rule:

- Independence assumption:
Building a Probabilistic Model

\[ P(M, W, F, T, S|H) = P(F)P(T, W|F)P(S|T, W, F) \times P(M|S, F, H) \]

- The sentence processing model is a model of 
  \( P(T, W, F, S) \). Maps \( W \) to \((F, S, T)\) triple (a 
  context-independent interpretation)

- The contextual processing model goes from a \((F, S, H)\) 
  triple to a final interpretation, \( M \)
### Example

<table>
<thead>
<tr>
<th>H=</th>
<th>Show: (flights)</th>
<th>Origin: (City &quot;NY&quot;) or (City &quot;NY&quot;)</th>
<th>Destination: (City &quot;Atlanta&quot;)</th>
<th>Date: (November 27th, 2003)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E,S=</th>
<th>Show: (Arrival-time)</th>
<th>Origin: (City “Newark”)</th>
<th>Destination: (City &quot;Atlanta&quot;)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>M=</th>
<th>Show: (Arrival-time)</th>
<th>Origin: (City “Newark”)</th>
<th>Destination: (City &quot;Atlanta&quot;)</th>
<th>Date: (November 27th, 2003)</th>
</tr>
</thead>
</table>
Building a Probabilistic Model

\[
\]

- First step: choose the frame \( F \) with probability \( P(F) \)

<table>
<thead>
<tr>
<th>Show:</th>
<th>(Arrival-time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td></td>
</tr>
<tr>
<td>Destination:</td>
<td></td>
</tr>
</tbody>
</table>
The Sentence Processing Model

\[ P(T, W, F, S) = P(F)P(T, W|F)P(S|T, W, F) \]

- Next step: generate the parse tree \( T \) and sentence \( W \)
- Method uses a probabilistic context-free grammar, where markov processes are used to generate rules. Different rule parameters are used for each value of \( F \)
The Sentence Processing Model

P(/det flight/corenp flight−constraints/rel−clause|flight/np)
= P(/det|NULL, flight/np) *P(flight/corenp|/det,flight/np)
* P(flight−constraints|relclause|flight/corenp,flight/np)
* P(STOP|flight−constraints/relclause,flight/np)

• Use maximum likelihood estimation

\[ P_{ML}(corenp|np) = \frac{\text{Count}(corenp, np)}{\text{Count}(np)} \]

• Backed-off estimates generate semantic, syntactic parts of each label separately
The Sentence Processing Model

- Given a frame $F$, and a tree $T$, fill in the semantic slots $S$

<table>
<thead>
<tr>
<th>Show:</th>
<th>(Arrival-time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>Newark</td>
</tr>
<tr>
<td>Destination:</td>
<td>Atlanta</td>
</tr>
</tbody>
</table>

- Method works by considering each node of the parse tree $T$, and applying probabilities $P(\text{slot-fill-action}|S,\text{node})$
The Sentence Processing Model: Search

\[
\]

- **Goal:** produce \( n \) high probability \((F, S, T, W)\) tuples

- **Method:**
  - In first pass, produce \( n \)-best parses under a parsing model that is independent of \( F \)
  - For each tree \( T \), for each possible frame \( F \), create a \((W, T, F)\) triple with probability \( P(T, W, |F) \). Keep the top \( n \) most probable triples.
  - For each triple, use beam search to generate several high probability \((W, T, F, S)\) tuples. Keep the top \( n \) most probable.
The Contextual Model

\[
\text{H=} \begin{align*}
\text{Show:} & \quad \text{(flights)} \\
\text{Origin} & \quad \text{(City ”NY“) or (City ”NY“)} \\
\text{Destination:} & \quad \text{(City ”Atlanta“)} \\
\text{Date:} & \quad \text{(November 27th, 2003)}
\end{align*}
\]

\[
\text{E,S=} \begin{align*}
\text{Show:} & \quad \text{(Arrival-time)} \\
\text{Origin} & \quad \text{(City “Newark“)} \\
\text{Destination:} & \quad \text{(City ”Atlanta“)}
\end{align*}
\]

\[
\text{M=} \begin{align*}
\text{Show:} & \quad \text{(Arrival-time)} \\
\text{Origin} & \quad \text{(City “Newark“)} \\
\text{Destination:} & \quad \text{(City ”Atlanta“)} \\
\text{Date:} & \quad \text{(November 27th, 2003)}
\end{align*}
\]
The Contextual Model

- Only issue is whether values in $H$, but not in $(F, S)$, should be carried over to $M$.

<table>
<thead>
<tr>
<th>M=</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show:</strong></td>
<td>(Arrival-time)</td>
</tr>
<tr>
<td><strong>Origin</strong></td>
<td>(City “Newark”)</td>
</tr>
<tr>
<td><strong>Destination:</strong></td>
<td>(City ”Atlanta”)</td>
</tr>
<tr>
<td><strong>Date:</strong></td>
<td>(November 27th, 2003)</td>
</tr>
</tbody>
</table>

- Method uses a decision-tree model to estimate probability of ”carrying over” each slot in $H$ which is not in $F, S$. 
Summary

• HMM model for DA labeling
• Architecture of Dialogue Systems
• Statistical model for the NLU component
• Next time:
  – Planning Agents
  – Markov Decision Processes for Dialogue Management