Speech Processing 11-492/18-492

Spoken Dialog Systems
SDS components
More than just ASR and TTS

- Recognition
- Language understanding
- Manipulation of utterances
- Generation of new information
- Text generation
- Synthesis
SDS Architecture

- ASR
- Language Understanding
- Language Generation
- Synthesis
- Dialog Manager

Error Handling Strategies
Language Understanding
- From words to structure

Dialog Manager
- State of dialog (who is talking)
- Direction of dialog (what next)
- References, user profile etc
- Interaction of database/internet

Language Generation
- From structure to words
 Parsing of SPEECH not TEXT

- Eh, I wanna go, wanna go to Boston tomorrow
- If its not too much trouble I’d be very grateful if one might be able to aid me in arranging my travel arrangements to Boston, Logan airport, at sometime tomorrow morning, thank you.
- Boston, tomorrow
“I wanna go to Boston, tomorrow”

- Destination: BOS
- Departure: 20081028, AM
- Airline: unspecified
- Special: unspecified

Convert speech to structure

- Sufficient for further processing/query
User: find a cheap eating place for Taiwanese food

Intelligent Agent:
Cheap Taiwanese eating places include Din Tai Fung, Boiling Point, etc. What do you want to choose? I can help you go there.
User: find a cheap eating place for taiwanese food

Intelligent Agent: seeking

- price (AMOD)
- food (NN)
- target

- seeking (PREP_FOR)
User

find a cheap eating place for taiwanese food

SDS Process

Ontology Induction (semantic slot)

Intelligent Agent

Organized Domain Knowledge
find a cheap eating place for taiwanese food
User

Intelligent Agent

find a cheap eating place for taiwanese food

price

AMOD

food

NN

seeking=

PREP_FOR

target=

seeking=“find”

target=“eating place”

food=“cheap”

food=“taiwanese”
User

Intelligent Agent

find a cheap eating place for taiwanese food

seeking

price

AMOD

food

target

PREP_FOR

Semantic Decoding

seeking=“find”
target=“eating place”
price=“cheap”
food=“taiwanese”
can i have a cheap restaurant

General
Frame: capability

Frame: expensiveness

Frame: locale by use

slot candidate

Domain

Domain
**Parsing vs Language Model**

- **Language Model**
  - Model what actually gets said

- **Parsing**
  - Extract the information you want

- **Models *can* be shared**
  - Only accept things in the grammar
  - Can be over limiting
RNN for Slot Filling

Step 1: word embedding
Step 2: short-term dependencies capturing
Step 3: long-term dependencies capturing
Step 4: different types of neural architecture

Mesnil et al. 2013

<table>
<thead>
<tr>
<th>Input (words)</th>
<th>show</th>
<th>flights</th>
<th>from</th>
<th>Boston</th>
<th>to</th>
<th>New</th>
<th>York</th>
<th>today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output (labels)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>B-dept</td>
<td>O</td>
<td>B-arr</td>
<td>l-arr</td>
<td>B-date</td>
</tr>
</tbody>
</table>
Luis: Interactive machine learning for language understanding

Advantages:
- Non-expert could add in knowledge in feature engineering
- Active-learning reduces heavy labeling

https://www.luis.ai/
Dialog Manager

- **Maintain state**
  - *Where are we in the dialog*
  - *Whose turn is it*
    - Waiting for speaker
    - Waiting for database query (stall user)
  - *Deal with barge-in*
Frame Based Dialog Manager

- Used for transaction dialog
- Generalizes finite-state approach by allowing multiple paths to acquire info
  - Central data structure is frame with slots – DM is monitoring frame, filling in slots
  - Frame:
    - Set of information needed
    - Context for utterance interpretation
    - Context for dialogue progress
- Allows mixed initiative
  - Allows over-answering
- Also called form-based (MIT): Often called “slot-filling”
Problems with Frames

- Not easily applicable to complex tasks
  - May not be a single frame
  - Dynamic construction of information
  - User access to “product”
Agenda + Frame

◆ **Product:**
  - hierarchical composition of frames

◆ **Process:**
  - Agenda
    - Generalization of stack
    - Ordered list of topics
    - List of handlers
Statistical Approaches to DM

- Allow for dialog complexity beyond human mind
- Find optimal decision for non-trivial design problems
- Life-long learning
Decisions

- **Difficult design decision over the course of interaction**
  - When to ask open / directive questions?
  - When to confirm?
  - When to barge-in / wait?
  - Which type of feedback to provide? (ex, intelligent tutoring system)

- **Sample efficient policy search**
  - Policy space is too huge to search with traditional ways of SDS development
S1: Welcome to CMU Let’s Go. Where do you leave from?
U1: CMU
S2: From CMU, did I get that right?
U2: Yes.
S3: Where are you going?
U3: Downtown.
S4: To Downtown, did I get that right?
U4: Yes.

---

S1: Welcome to CMU Let’s Go. How may I help you?
U1: I'd like to go from CMU to Downtown.
S2: From CMU to Downtown, did I get that right?
U2: Yes.
S3: When are you going to take the bus?
U3: Now
S3: You want the next bus, is that right?
U3: Yes.
S1: Welcome to CMU Let’s Go. Where do you leave from?
U1: CMU
S2: From CMU, did I get that right?
U2: Yes.
S3: Where are you going?
U3: Downtown.
S4: To Downtown, did I get that right?
U4: Yes.

S1: Welcome to CMU Let’s Go. Where do you leave from?
U1: CMU • S2: Where are you going?
U2: Downtown.
S3: When are you going to take bus?
U3: Now
S4: Leaving from CMU going to Downtown immediately, is it correct?
Explicit Confirm VS Implicit Confirm

S1: Welcome to CMU Let’s Go. Where do you leave from?
U1: CMU
S2: From CMU, did I get that right?
U2: Yes.
S3: Where are you going?
U3: Downtown.
S4: To Downtown, did I get that right?
U4: Yes.

S1: Welcome to CMU Let’s Go. Where do you leave from?
U1: CMU
S2: Departing CMU. Where are you going?
U2: Downtown.
S3: Going to Downtown. When are you going to take bus?
U3: Now
Corrective Feedback

S1: Welcome to CMU Let’s Go. How may I help you?
U1: I likes go CMU to Downtown.

S2: You must not use third person singular verb after first person singular pronoun [explicit correction]

S1: Welcome to CMU Let’s Go. How may I help you?
U1: I likes go CMU to Downtown.

S2: You said I like to go from CMU to Downtown. Did I get that right? [recast]
Traditional Policy Design

- **Traditional SDS development paradigm**
  - Several versions of a system are developed (where each version uses a single dialog policy, intuitively designed by an expert)
  - Dialog corpora are collected with human users interacting with different versions of the system
  - A number of evaluation metrics are measured for each dialog
  - The different versions are statistically compared
  - Update the system with “best” dialog policy

- **Due to the costs of experimentation, only a handful of policies are usually explored in any one experiment**
The number of possible policies is much larger

Limited Data
Agent updates state by observing environment
Given the state, agent performs the best action to achieve its goal
Need to learn the mapping $h$ from $s \in S$ to $a \in A$
Supervised learning

- For each input state $s$, optimal action $a$ is known
- Infer input-output relationship $a \approx h(s)$
- Example: neural networks, support vector machine
Markov Decision Process

- No data:
  - Human-Human is differences between human and machine perception
  - Wizard of Oz is costly and wizard is hard to train.

- Statistical method to formulate the question

  MDP computes an optimal dialog policy within a much larger search space, using a relatively small number of training dialog
  MDP evaluates actions (small number) not policies (large number) by credit assignment of total reward to each action using dynamic programming
Markov Decision Process

- Transition function $T(s, a, s')$ defines environment behavior
- Reward function $r(s, a, s')$ defines immediate reward of transition to $s'$ by performing $a$ in state $s$
- Return $R^h(s_t)$ defines discounted total rewards:
  - $R^h(s_t) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, h(s_k), s_{k+1})$
Learning Goal

Find $h$ that maximizes discounted return

$$R^h(s_t) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, h(s_k), s_{k+1})$$

from any $s_0$

- Discount factor $\gamma \in [0; 1)$:
  - Encodes increasing uncertainty about the future
  - Bounds infinite sum
  - Helps convergence of algorithms
Dialog Management as MDP

- What a dialog looks like
  - System says something
  - User says something
  - System says something
  - User says something
  - System says something
  - User says something
  - ...

- We can think of a dialog as a trajectory in state space

\[ s_0, a_0, r_0 \rightarrow s_1, a_1, r_1 \rightarrow s_2, a_2, r_2 \rightarrow \ldots \]

- The best policy is the one with the greatest expected reward over all trajectories
Model Dialog

- What representation of the dialog state best serves this setting?
- What actions are needed?
- Where does transition function come from?
- What reward functions are needed?
State

- MDP assumes that state transitions are Markovian
  \[ P(s_{t+1} \mid s_t, s_{t-1}, ..., s_o, a_t, a_{t-1}, ..., a_o) = P_T(s_{t+1} \mid s_t, a_t) \]

- In principle, MDP state could include any possible information about dialog
  - Complete dialog 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
Actions

- Dialog acts
  - Ask a question
  - Explicit confirmation
  - Rejection
  - Give the user some database information
  - Tell the user their choices

- Do a database query
Transition Functions

• If we had labeled training data

\[ P(s'|s, a) = \frac{C(s, s', a)}{C(s, a)} \]

• How to estimate \( p(s'|s, a) \) without labeled data
  • Random conversations with real people
    • Carefully hand-tune small number of states and policies
    • Build a dialog system which explores state space by generating a few hundred random conversations with real humans
  • Set probabilities from this corpus
  • Random conversations with simulated users
    • Have millions of conversations with simulated users
    • Can have a slightly larger state space
Reward Function

• If we knew the final reward for whole dialog, we can learn Q values (pushing back reward values over state sequences) and hence best policy

• What is the final reward for whole dialog?
  • Task success / failure
  • General goodness of a whole dialog that can be automatically computed using PARADISE evaluation framework

• Small negative reward for all non-terminal states
  • Encourage more efficient dialog
Example: Non-Task-Oriented System Architecture

- User Input
  - Context Tracking Strategies
  - Engagement Appropriateness Strategies
  - High Generation Confidence
  - Trigger Condition Meet
- System Output
  - Response Generation Method
  - Lexical Semantic Strategies
  - Low Generation Confidence
  - Trigger Condition Not Meet
- Response Generation
  - Conversation Strategy Selection
  - Trigger Condition Meet
Active Participation

Start conversations with a concrete topic
  e.g. TickTock: Hi, I am TickTock, I really like sports, let’s talk about sports.

Design each strategies with respect to a topic
  Stay on the topic
  Change the topic
Stay on the Topic:

S1: Tell a joke (joke)
   e.g. Do you know that people usually spend far more time watching sports than actually playing any.

S2: Initiate activities (initiate)
   e.g. Do you want to see a game together some time?

S3: Talk more (more):
   e.g. Let’s talk more about sports.
S4: Switch topics (switch):
   e.g. How about we talk about movies?

S5: End topics with an open question (end):
   e.g. That’s interesting, could you share with me some interesting news on the Internet?
TickTock: Hello, I really like politics. Let’s talk about politics.

User: No, I don’t like politics.

TickTock: Why is that?

User: I just don’t like politics.

TickTock: How about we watch a debate together some day? (initiate activity)

User: No, you I don’t want to talk about politics!
TickTock: Hello, I really like politics. Let’s talk about politics. neutral
User: No, I don’t like politics. negative
TickTock: Why is that? neutral
User: I just don’t like politics. negative
TickTock: OK, how about we talk about movies? (switch topics)
User: I do watch movies a lot. neutral

Appropriate!
Engagement Maintain Policy

Goal: Improve system **appropriateness** considering context. \((S,A,R,\gamma,\alpha)\)

Method: Reinforcement learning

Q Learning : \(Q:S \times A \rightarrow R\)

**State Variable (S):**
- system-appropriateness confidence
- all previous utterance-sentiment confidence
- time of each strategy executed
- turn position
- most recently used strategy

**Actions (A) :** engagement strategies (5 types) and generated utterance
Reward function \( R \): Cumulated Appropriateness \(*0.3\) + Conversation depth\(*0.3\) + Information gain \(*0.4\)

**Appropriateness**: the current response’s coherence with the user utterance.

Automatic predictor: SVM binary classifier (Inappropriate, interpretable VS Appropriate)

**Conversation depth**: the maximum number of consecutive utterances on the same topic.

Automatic predictor: SVM binary classifier (Shallow, Intermediate VS Deep)

**Information gain**: the number of unique tokens.

Update: \[ Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha(s_t, a_t) \cdot (R_{t+1} + \gamma \max_a Q_t(s_{t+1}, a_t) - Q_t(s_t, a_t)) \]

Simulator: A.L.I.C.E. chatbot. Interface: Multithreaded text-only API (open source)
Query for flights to Boston

Template fill answer(s)

- The next flight to DEST leaves at DEPART_TIME arriving at ARRIVE_TIME.

Templates may be much more complex
Language Generation

- Choose which template to use
  - Based on state, answer type
  - Natural variation
  - Statistical variation

- Include `<ssml>` tags to help synthesis
  - Can `<emph>`emphasize`</emph>` parts
  - Can identify dates, numbers etc.

- Humans like variation in the output
  - It is rare for a human to repeat things exactly
Frames structures to (marked up) text

- START: Pittsburgh
- END: Boston
- DATE: 20081028
- TIME: 07:45
- FLIGHT: US075

Can generate

- I have US 075 leaving at 07:45 tomorrow
- US Airways has a flight departing tomorrow at 07:45
Standardized things

Help

- User should be able to get help at any time
- Explain where they are and what they are expected to say (with explicit examples)

Errors

- “I didn’t understand” …

Confirmation

- Did you say “Boston”?
Explicit confirmation

- Where are you traveling to?
  Boston

- Boston, did I get that right?
  Yes
Implicit confirmation

- Where are you traveling to?
  Boston

- Boston, where …
  <can barge in>
Explicit confirmation
- Safe but slow

Implicit confirmation
- Natural, but requires good support for barge-in
**Grounding**

- **Showing evidence the system understands**
  - Where are you traveling to?
    - Boston.

  Right. Where ....

  Boston, right. Where ....
Constrain your questions:

- How may I help you?
  - Long story reply
- What bus number would like schedules for?
  - Expect bus number replies