CSCI 544, Lecture 8: Dialogue

Ron Artstein

2022-09-15

Introducing myself

Ron Artstein

- Research Scientist at ICT
- Natural Language Dialogue
- Linguist by training
- Teaching this course since 2016

Applied NLP

- Until 2013, this course was simply called “NLP”.
- Concentrate on NLP applications.
- Learn to think about the language in the problem.

Administrative notes

Written Assignment due tonight

Written Assignment Peer Grading starts tomorrow, due September 22

Coding Assignment 2 due September 27

Project:

Due Date	Task
September 20	Form project teams (52 teams)
September 20–29	Initial discussion with assigned TA
October 4	**Project proposal**
November 3	Project status report
Nov 29/Dec 1	Poster presentations (in class)
December 1	Final report
December 3	Self-evaluation and peer grading

More administrative notes

Use **Piazza** for questions

Make Piazza questions **public** whenever possible

- Other students can answer
- Other students can see the answer

This is a big class; many students have similar issues

Ambiguity and an NLP application

> — Hey Siri, tell Nancy David wants spaghetti.
>
> — I'm sorry, I couldn't find a Nancy David in your contacts.

Observations:

- The intended contact was "Nancy".
- Local ambiguity resolved by the end of the utterance:

  Tell Nancy David...

We can guess how Siri works:

Tell	Nancy David	wants spaghetti
↓	↓	↓
instr	contact	msg content

- Parser does not see my contacts
- Parser might not consider grammaticality of full utterance.

> ☛ Many design considerations
>
> ☛ Think of the design for your project!

Dialogue systems

Dialogue systems are very common these days

- Phones: Siri (2011), Cortana (2015–2021), Google (2016), ...
- Speakers: Alexa (2014), Google (2016), ...

Much older history

- Spoken: >30 years
- Text-based: >50 years (Eliza)

Can be topic of a whole course

- ☛ CSCI 644 Natural Language Dialogue Systems

Structure of today's lecture

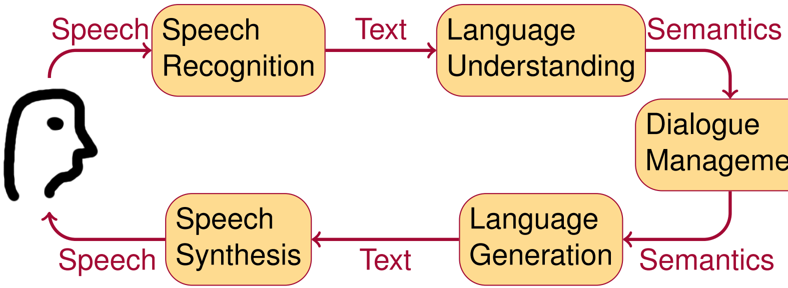
1. Dialogue system architecture
2. Dialogue management for task-based systems
3. Conversational systems
4. Evaluation of dialogue systems

Task/goal-based:	Specific, external task
Conversational:	Sustain conversation

Most systems have some of both elements

Conversational goals (even without external goal)

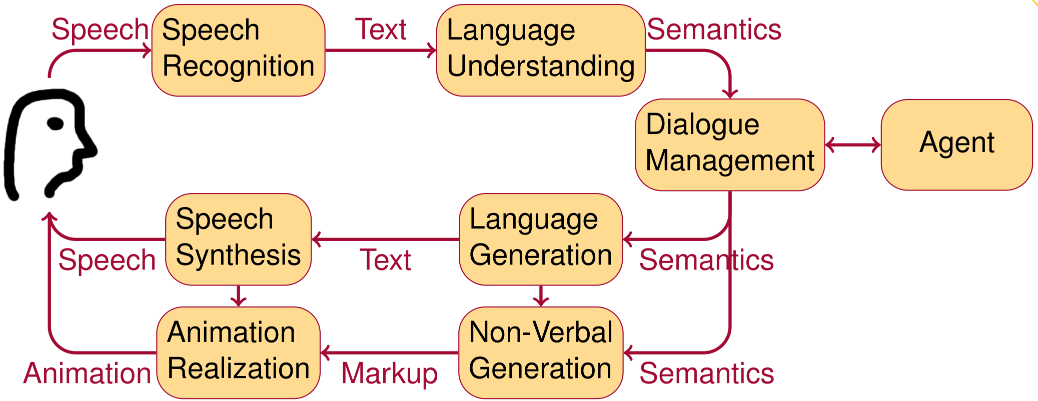
Spoken dialogue systems: prototypical architecture



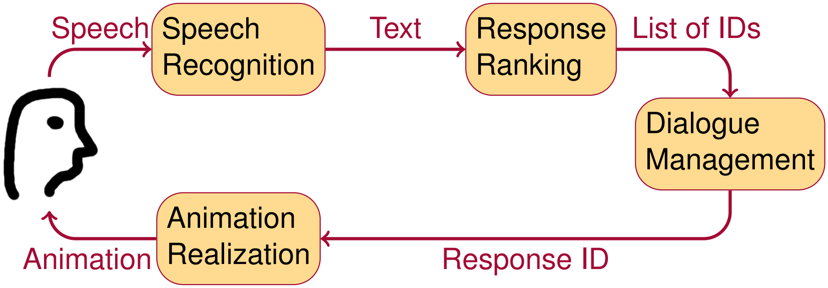
Information flow, not pipeline

- Input does not necessarily result in output

Spoken dialogue systems: additional components



Spoken dialogue systems: fewer components



Dialogue management

Policy/Strategy Action to take at each point in the dialogue

- Requires taxonomy of actions
- May relate to user input (but not only)

Dialogue State Where the conversation is now

- A set of state variables

Example Policy Always maximize expected utility

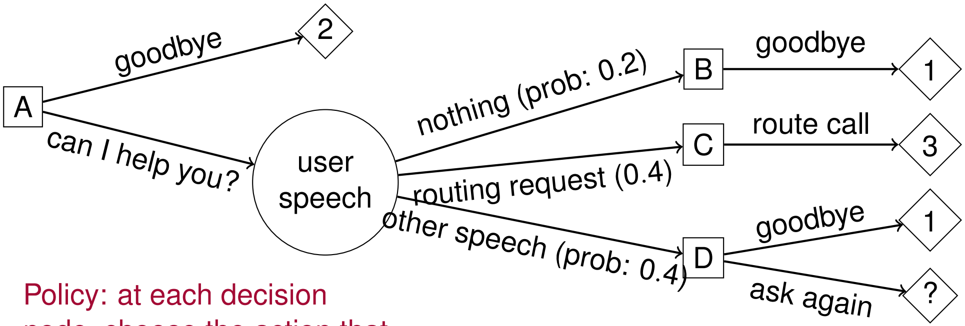
- Need to calculate expected utility

Dialogue acts

Assertion
Question
Request
Command
Promise
Threat
etc.

Decision trees

□ Action node ○ Observation node ◇ Reward node



Policy: at each decision node, choose the action that maximizes expected utility

Markov Decision Process

Sets of states S, actions A, and rewards R (or costs)

- Probabilistic transition between states
- Probabilistic rewards
- Actions are chosen by the agent

Markov property for MDPs

- Probability of transition, reward depends only on the current state and action

$$\begin{aligned} P(s_{t+1}|\text{history}) &\approx P(s_{t+1}|s_t, a_t) \\ P(r_{t+1}|\text{history}) &\approx P(r_{t+1}|s_t, a_t) \end{aligned}$$

Rewards can also be fixed (for each state-action pair)

MDP policies

Policy What action to take at each point in the dialogue

- A function from states to actions: $\pi : S \rightarrow A$

Optimal policy $\pi : S \rightarrow A$ which maximizes expected reward

- = minimizes expected cost

MDP model assumes the system always knows what state it is in

Partially Observable MDP

Add a set of observations O and emission probabilities

- The probability of an observation depends only on the current state and action

$$P(o_{t+1} | \text{history}) \approx P(o_{t+1} | s_{t+1}, a_t)$$

Belief (at time t) = distribution over states

Learning (PO)MDPs

Need to learn: Transition probabilities
Observation probabilities
Immediate rewards (or reward probabilities)

Can we learn from a corpus?

- Corpus reflects the policy that produced the corpus

Reinforcement learning: learn from interaction

- update probabilities, rewards based on interaction
- Reward good outcomes, penalize bad outcomes
- Propagate rewards backwards

Requires a lot of interaction

Simulated users

User simulation

- Simulate actions
- Simulate utterances (including ASR and other errors)

Simulated users often learned from a corpus

When rewards are unknown: Inverse reinforcement learning

- Learn the reward function that results in a policy that mimics expert users

Dialogue state tracking

Dialogue State = where the conversation is now

- A set of state variables
- For task-oriented: which slots have been filled

Dialogue System Technology ~~State Tracking~~ Challenge

Relatively well-defined for simple tasks

Complex tasks might

- share slot values between tasks
- express complex goals in a single utterance
- interleave related tasks

Task lineages: framework

Lee and Stent (2016). Task Lineages: Dialog State Tracking for Flexible Interaction. Sigdial.

Multi-task dialogue (slot-filling)

What is a task? — Book a restaurant, book a ride, ...

- Each task requires some information in order to be executed

Task schema required and optional slots for operationalization

Dialogue act item slot + value

Task frame parse tasks + associated DAI + confidence + $\cdots$

Task lineage history of states; maintain in parallel due to ambiguity

Update function extend, then prune (global)

Conversational dialogue systems

Emphasis on maintaining conversation

Additional goals: education, entertainment, connection

New Dimensions in Testimony demo

- Conversational question-answering character
- Primarily reactive (responds to user questions)

To be continued...