CS447: Natural Language Processing

http://courses.engr.illinois.edu/cs447

Lecture 26: Dialogue

Julia Hockenmaier

juliahmr@illinois.edu

3324 Siebel Center

The linguistics of dialogue

Today's lecture

The linguistics of dialogue
What do we know about what happens
when two or more people are having a conversation?

Dialogue Systems/Conversational Agents

How can we design systems to have a conversation with a human user?

- **Chatbots**
  Mostly chitchat, although also some use in therapy
- **Task-based Dialogue Systems**
  Help human user to accomplish a task
  (e.g. book a ticket, get customer service, etc.)

Recap: Discourse

On Monday, John went to Einstein's. He wanted to buy lunch. But the cafe was closed. That made him angry, so the next day he went to Green Street instead.

'Discourse':
Any linguistic unit that consists of multiple sentences

Speakers describe "some situation or state of the real or some hypothetical world" (Webber, 1983)

Speakers attempt to get the **listener** to construct a similar **model of the situation**.

A dialogue between a customer (C) and a travel agent (A)

C_1: ...I need to travel in May.
A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.
A_3: And what time would you like to leave Pittsburgh?
C_4: Uh hmm I don't think there's many options for non-stop.
A_4: Right. There's three non-stops today.
C_5: What are they?
A_5: 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.
C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_7: OK.
A_7: And you said returning on May 15th?
C_8: Uh, yeah, at the end of the day.
A_8: OK. There's #two non-stops . . . #
C_9: #Act. . . actually #, what day of the week is the 15th? A_9: It's a Friday.
C_{10}: Uh hmm. I would consider staying there an extra day til Sunday.
A_{10}: OK...OK. On Sunday I have ...

A dialogue between a customer (C) and a travel agent (A)

C_1: ...I need to travel in May.
A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.

A **dialogue** is a conversation between two speakers that consists of a sequence of **turns**

Turn = an **utterance** by one of the two speakers

Turn-taking requires the ability to detect when the other speaker has finished

Multiparty dialogue: A conversation among more than two speakers

From discourse to dialogue

Discourse:
The **speaker** communicates to an **absent, passive listener** (or audience), and attempts to get them to construct a similar model of the state of affairs.

The speaker does not receive any feedback from the audience.

Dialogue:
Both parties are **present** and **active** participants.
They each bring **their own mental model** of the state of affairs.
Communication succeeds if both parties understand each other's mental models (and perhaps even get their models to agree).

Both parties provide feedback to each other.

Grounding in Dialogue

For communication to be successful, both parties have to know that they understand each other (or where they misunderstand each other)

— Both parties maintain (and communicate) ***their own beliefs about the state of affairs*** that they're talking about.

— Both parties also maintain ***beliefs about the other party's beliefs about the state of affairs*.**

— Both parties also maintain ***beliefs about the other party's beliefs about their own beliefs***,… etc.

Common ground: The set of **mutually agreed beliefs** among the parties in a dialogue

Grounding in Dialogue

John: *Dragons are scary!*

Common Ground: {"*John thinks dragons exist*",
"*Mary knows that John thinks dragons exist*",
"*John finds dragons scary*"
"*Mary knows that John finds dragons scary*", ….}

If Mary replies: *What dragons?*

—> **Additions to Common Ground:**
{"*Mary doesn't think dragons exist*",
"*John knows that Mary doesn't think dragons exist*", …}

If Mary replies instead: *No, dragons are cute!*

—> **Additions to Common Ground:**
{"*Mary and John both think dragons exist*",
"*Mary finds dragons cute.*"
"*John knows that Mary finds dragons cute*",
"*Mary disagrees with John that dragons are scary*",…}

Clark and Schaefer: Grounding

When two people communicate, they each contribute to the conversation by establishing and adding to the common ground.

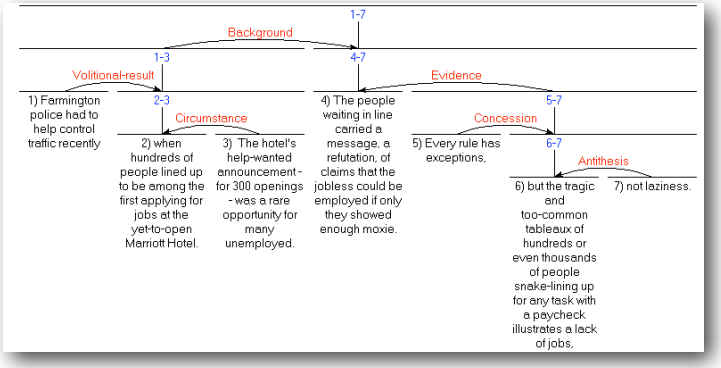
B needs to provide evidence of understanding what A meant by

- **Continued attention:** B continues attending to A
- **A relevant next contribution:** ("so, what shall we do now?")
- **Acknowledgement:** B nods, says (uh-huh, yeah), gives assessment (great!)
- **Demonstration of understanding:** B paraphrases or reformulates A's contribution, or completes A's utterance
- **Display:** B repeats verbatim all or part of A's utterance

Without such evidence, A may not know if B understood what A meant to convey.

Discourse has structure

Rhetorical relations hold between parts of a discourse (e.g. as formulated in RST, or the PDTB):



Different parts of a discourse play **different functions** (e.g. as in Argumentative Zoning for scientific papers):

Category	Description	Example
AIM	Statement of specific research goal, or hypothesis of current paper	"The aim of this process is to examine the role that training plays in the tagging process"
OWN_METHOD	New Knowledge claim, own work: methods	"In order for it to be useful for our purposes, the following extensions must be made:"
OWN_RESULTS	Measurable/objective outcome of own work	"All the curves have a generally upward trend but always lie far below backoff (51% error rate)"

Dialogues have structure too

A0: Hello, how can I help you?	Greeting, Question
C0: Hi, thanks! I'd like to book a flight.	Greeting, Answer
C_1: ...I need to travel in May.	Constraint

A_1: And, what day in May did you want to travel?
C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_2: And you're flying into what city?
C_3: Seattle.
A_3: And what time would you like to leave Pittsburgh?
C_4: Uh hmm I don't think there's many options for non-stop.
A_4: Right. There's three non-stops today.
C_5: What are they?
A_5: 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.
C_6: OK I'll take the 5ish flight on the night before on the 11th.
A_6: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_7: OK.
A_7: And you said returning on May 15th?
C_8: Uh, yeah, at the end of the day.
A_8: OK. There's #two non-stops . . . #
C_9: #Act. . . actually #, what day of the week is the 15th? A_9: It's a Friday.

Dialogues have structure too

Dialogues have (hierarchical) structure:

"Adjacency pairs": Some acts (first pair part) typically followed by (set up expectation for) another (second pair part):

Question → Answer,
Proposal → Acceptance/Rejection, etc.

Sometimes, a **subdialogue** is required (e.g. for clarification questions):

A: I want to book a ticket for tomorrow

B: Sorry, I didn't catch where you want to go?
A: To Chicago
B: And where do you want to leave from?
…

B: Okay, I've got the following options: …

Speech Acts

Utterances correspond to **actions** by the speaker, e.g.

- **Constative** (answer, claim, confirm, deny, disagree, state)
  Speaker commits to something being the case
- **Directive** (advise, ask, forbid, invite, order, request)
  Speaker attempts to get the listener to do something
- **Commissive** (promise, plan, bet, oppose)
  Speaker commits to a future course of action
- **Acknowledgment** (apologize, greet, thank, accept apology)
  Speaker expresses an attitude re. listener with respect to a social action

In practice, much more fine-grained labels are often used, e.g:

Yes-No Questions, Wh-Questions, Rhetorical Questions, Greetings, Thanks,
Yes-Answers, No-Answers, Agreements, Disagreements, …
Statements, Opinions, Hedges, …

Initiative

Who controls the conversation?

— Who asks questions?

— Who introduces new topics?

Human-human dialogue is typically **mixed initiative** where both parties take initiative at different points

(But it is difficult to design mixed initiative dialogue systems)

Systems often assume a **user-initiative strategy**

(User asks questions, System responds)

or a **system-initiative strategy**

(Only system can ask questions.
System-initiative systems can be very frustrating to use)

Inference and implicature

> A_1: And, what day in May did you want to travel?
> C_2: OK uh I need to be there for a meeting that's from the 12th to the 15th.

The customer (speaker) doesn't answer the question directly, but assumes the provided information allows the agent (hearer) to infer the requested information
(→ customer needs to travel on 11th/12th—15th/16th)

Hearers can draw these inferences ("**conversational implicatures**") because they assume speakers are being cooperative

Gricean maxims: rules that govern (cooperative) human communication. Here: "Be relevant!"

Chatbots

Chatbots vs Dialogue Systems

Chatbots: Chitchat, often used for entertainment, originally as testbed for clinical therapy

Dialogue Systems: Typically to perform specific tasks (e.g. customer service, reservations, etc., smart devices, cars, etc.)

A computer therapist?

Computer: TELL ME MORE ABOUT YOUR FAMILY

Human: My mother takes care of me.

Computer: WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU

Human: My father.

Computer: YOUR FATHER

Human: You are like my father in some ways.

Computer: WHAT RESEMBLANCE DO YOU SEE

Human: You are not very aggressive but I think you don't want me to notice that.

Computer: WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE

Human: You don't argue with me.

Computer: WHY DO YOU THINK I DON'T ARGUE WITH YOU

Human: You are afraid of me.

Weizenbaum (1966), ELIZA.

ELIZA as a FST cascade

Human: *You don't argue with me.*
Computer: *WHY DO YOU THINK I DON'T ARGUE WITH YOU*

1. Replace **you** with ***I*** and ***me*** with ***you:***
   I don't argue with you.
2. Replace ***<...>*** with ***Why do you think <...>:***
   Why do you think I don't argue with you.

Current Chatbots

Retrieval-based approaches:

Mine large corpora of human-human dialogues

Given a user utterance, find an utterance in the corpus that is most similar to this utterance.
Return either this utterance itself, or the response to it

Neural approaches (seq2seq models)

Trained on large corpora of human-human dialogues

May require beam search or other training objectives, and reinforcement learning or adversarial training to optimize for maintaining a longer conversation


Task-based
Dialogue Systems

Dialogue systems

Systems that are capable of performing
a **task-driven dialogue** with a human user.

Standard tasks:

- Travel arrangements
- Telephone call routing,
- Customer support
- Tutoring system

A travel dialog: Communicator

S1:	Hello. You've reached the [Communicator]. Tell me your full name
U2:	*hi i'd like to fly to seattle tuesday morning*
S3:	Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?
U4:	*[John Doe]*
S5:	Hello, [John], I'm glad to hear from you again. You can now ask me for help.
S6:	OK. Let's see I have a United flight departing Pittsburgh at ten forty five a.m. arriving into Seattle at one fifty seven p.m. Is that OK?
U7:	*Yes*
S8:	Will you return to Pittsburgh from Seattle?
U9:	*no i will go to Los Angeles*
S10:	Traveling to Los Angeles. on what date
U11:	*Sorry can we go back to talk about the previous leg again*
S12:	For the first leg, you are on a United flight 1446 departing Pittsburgh at ten forty five A.M. tomorrow, arriving in Seattle at one fifty seven P.M. Do you want to take that?
U13:	*actually i want to leave on wednesday*
S14:	a flight on wednesday.
S15:	I have a flight departing Pittsburgh at ten forty five a.m., arrives Seattle at one fifty seven p.m. Is that OK?
U16:	*Yes*

Figure 19.1 The travel domain: a fragment from a successful conversation between a user (U) and the Communicator system (S) of Xu and Rudnicky (2000).

Call routing: ATT HMIHY

S:	How may I help you?
U:	*can you tell me how much it is to tokyo?*
S:	You want to know the cost of a call?
U:	*yes that's right*
S:	Please hold on for rate information.

S:	How may I help you?
U:	*what's the area code for chicago?*
S:	You want an area code.
U:	*(silence)*
S:	The area code for Chicago is 312.

Figure 19.2 The call routing/telephone information domain: two actual conversations between a user (U) and the AT&T How May I Help You (HMIHY) system (S) (Gorin et al., 1997).

A tutorial dialogue: ITSPOKE

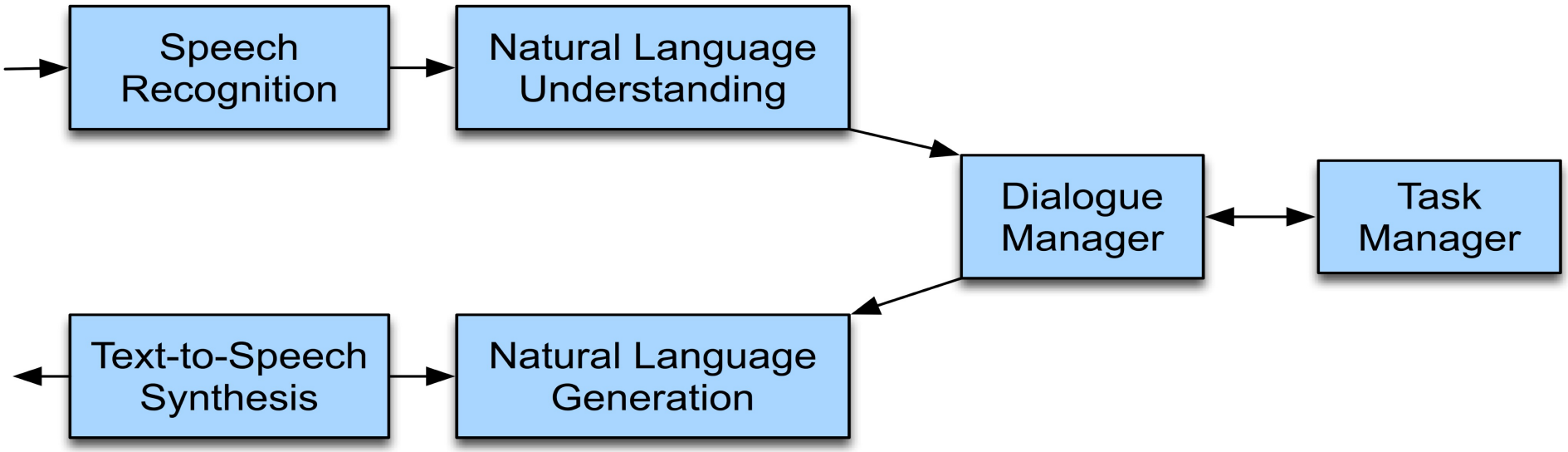
S: I didn't understand. What force acts on all objects within close proximity to earth?
U: *uh the force of gravity*
S: Fine. Besides the Earth's gravitational force, are there any other forces acting on the person, keys, and elevator after the release?
U: *no*
S: Yep. Thus, keep in mind that after the release, the only force on the person, keys, and elevator is the force of gravity. If gravity is the only force acting on an object, will it be moving or staying still?
U: *it will be moving*
S: Fine. Not only are the person, keys, and elevator moving, they have only gravitational forces on them. When an object is falling and has only gravitational force on it, it is said to be in what?
U: *freefall*

Figure 19.3 The tutoring domain: part of an actual conversation between a student user (U) and the ITSPOKE system (S) of (Litman and Silliman, 2004), based on the Why2-Atlas text-based tutoring system (?).

GUS: State of the art in 1977

GUS:	Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?
Client:	I want to go to San Diego on May 28
GUS:	What time do you want to leave?
Client:	I must be in San Diego before 10 am
GUS:	Would you like P.S.A. flight 102 that arrives at 9.15 am ?
Client:	What is the next flight ?
GUS:	Air California flight 310 that leaves at 8.30am and arrives at 10.00am
Client:	I'll take the first one
GUS:	In what name should I make a reservation, please?
Client:	Dan Bobrow
GUS:	I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.
GUS:	What date do you want to return on ?
Client:	On Friday in the evening.
GUS:	Would you like the flight that leaves at 7.45 pm ?
Client:	That's fine.
GUS:	I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

Dialogue System Architecture



Dialogue Manager

Controls the architecture and structure of dialogue

- Takes input from ASR (speech recognizer) and NLU components
- Maintains some sort of internal state
- Interfaces with Task Manager
- Passes output to Natural Language Generation/ Text-to-speech modules

Task-driven dialog as slot filling

If the purpose of the dialog is to complete a specific **task** (e.g. book a plane ticket), that task can often be represented as a **frame** with a number of **slots** to fill.

The task is completed if all necessary slots are filled.

This assumes a "**domain ontology**":

A knowledge structure representing possible user intentions for the given task

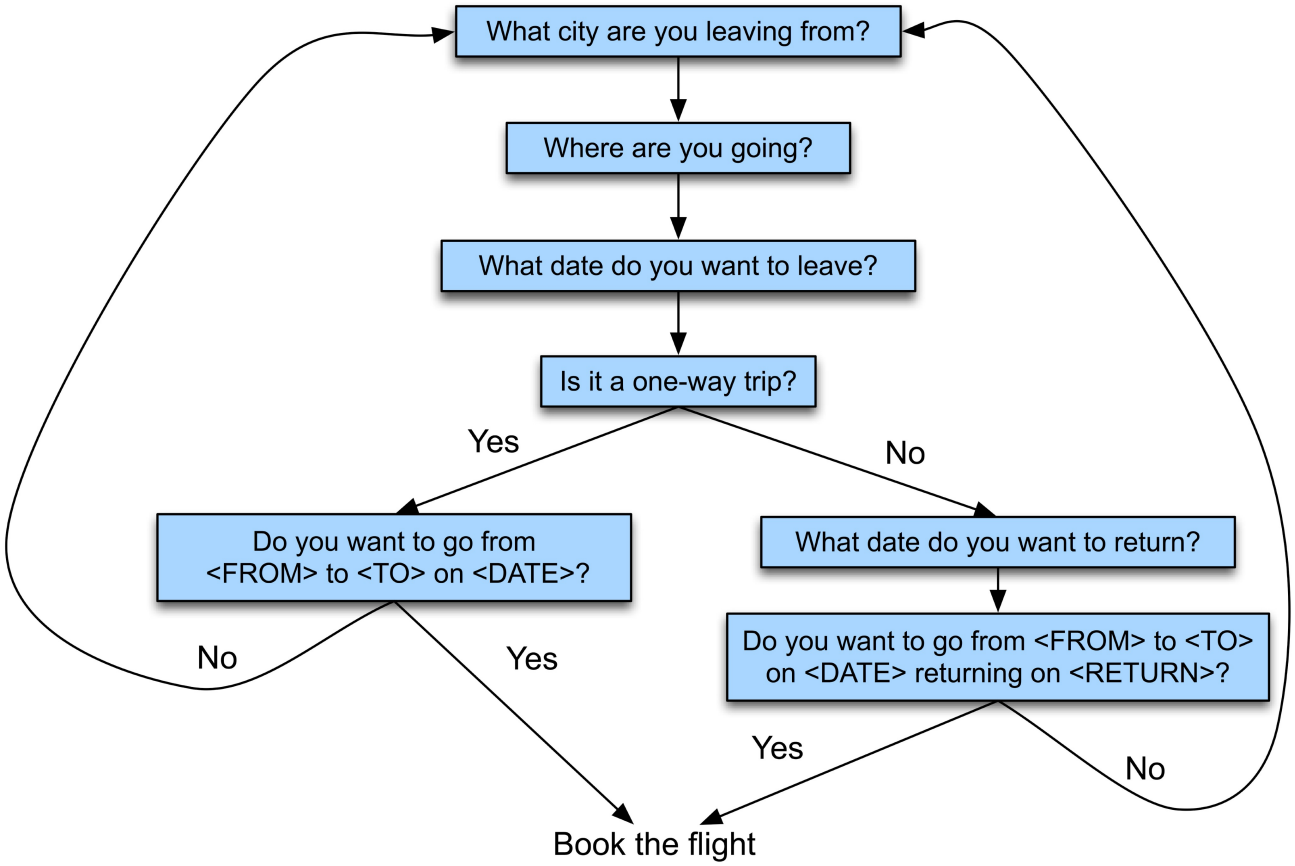
The Frame

A **frame** is set of **slots**, each to be
— filled with information of a given **type**, and
—associated with a **question** to the user

Slot	**Type**	**Question**
ORIGIN	city	*What city are you leaving from?*
DEST	city	*Where are you going?*
DEP-DATE	date	*What day would you like to leave?*
DEP-TIME	time	*What time would you like to leave?*
AIRLINE	line	*What is your preferred airline?*

Finite-state dialogue managers

Represent dialog structure as a finite state diagram



Purely sytem initiative

NLU with frame/slot semantics

But if we map user utterances to frames, we can detect which slots are filled or remain to be filled:

Show me morning flights from Boston to SF on Tuesday.

The system needs to identify the flight frame and fill in the correct slots:

```
SHOW:   FLIGHTS: ORIGIN: CITY: Boston
                         DATE: Tuesday
                         TIME: morning
                 DEST:   CITY: San Francisco
```

This allows for mixed-initiative dialogue systems.

Information-State and Dialogue Acts

If we want dialogue systems to allow the user to take the initiative, it needs to be able to…

…Decide when the user has asked a question, made a proposal, rejected a suggestion

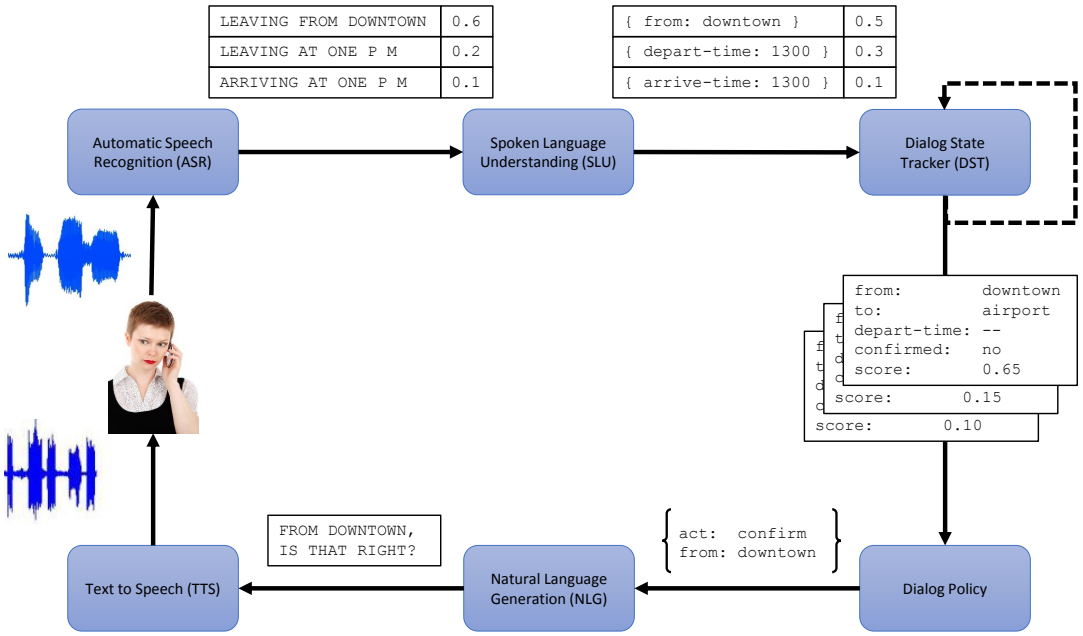
…Ground a user's utterance, ask clarification questions, make new suggestions

This implies that:

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 (even if the goal is still to fill a frame)

The Dialogue-State Architecture



Speech Recognition: Transcribe the spoken user utterance

Language Understanding: Extract slots from the utterance

Dialogue State Tracking: Maintain current state of dialogue

Dialogue Policy: What should the system say next?

Natural Language Generation: Produce an utterance to return.

Text-to-Speech: Convert the utterance to audio signal.

Dialogue Acts

Domain specific ontology,
consisting of

— **Dialog Act Labels (Intent)**
Functions of an utterance

— **Dialog Act Contents**
Slots that can be filled in an act

Tag	Sys	User	Description
HELLO($a = x, b = y, \ldots$)	✓	✓	Open a dialogue and give info $a = x, b = y, \ldots$
INFORM($a = x, b = y, \ldots$)	✓	✓	Give info $a = x, b = y, \ldots$
REQUEST($a, b = x, \ldots$)	✓	✓	Request value for a given $b = x, \ldots$
REQALTS($a = x, \ldots$)	χ	✓	Request alternative with $a = x, \ldots$
CONFIRM($a = x, b = y, \ldots$)	✓	✓	Explicitly confirm $a = x, b = y, \ldots$
CONFREQ($a = x, \ldots, d$)	✓	χ	Implicitly confirm $a = x, \ldots$ and request value of d
SELECT($a = x, a = y$)	✓	χ	Implicitly confirm $a = x, \ldots$ and request value of d
AFFIRM($a = x, b = y, \ldots$)	✓	✓	Affirm and give further info $a = x, b = y, \ldots$
NEGATE($a = x$)	χ	✓	Negate and give corrected value $a = x$
DENY($a = x$)	χ	✓	Deny that $a = x$
BYE()	✓	✓	Close a dialogue

	Utterance	Dialogue act
U:	Hi, I am looking for somewhere to eat.	`hello(task = find,type=restaurant)`
S:	You are looking for a restaurant. What type of food do you like?	`confreq(type = restaurant, food)`
U:	I'd like an Italian somewhere near the museum.	`inform(food = Italian, near=museum)`
S:	Roma is a nice Italian restaurant near the museum.	`inform(name = "Roma", type = restaurant, food = Italian, near = museum)`
U:	Is it reasonably priced?	`confirm(pricerange = moderate)`
S:	Yes, Roma is in the moderate price range.	`affirm(name = "Roma", pricerange = moderate)`
U:	What is the phone number?	`request(phone)`
S:	The number of Roma is 385456.	`inform(name = "Roma", phone = "385456")`
U:	Ok, thank you goodbye.	bye()

Dialogue annotated with dialogue acts and slot information

From Slots to Dialogue States

The **domain** of a slot-based dialogue system is defined by a set of **slots** and **possible slot values**, often given in a **database**.

The **goal** of a dialogue is to find one entity (e.g. a restaurant) in this database or to perform a specific action (e.g. booking a table at a restaurant at a particular time) that satisfies the **constraints** the user specifies

Restaurant domain (for recommendations, reservations,…)
Slots: Restaurant Name, Address, Cuisine, Price, Ratings,… [attributes]
Slot values: these attributes for any given restaurant

Informable slots:
Attributes that can be used to constrain the search
Requestable slots: Users can ask the value of these slots.
NB: Slots can be informable *and* requestable (cuisine, ratings), or just requestable (e.g. phone number)

Henderson, Machine Learning for Dialogue State Tracking.
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44018.pdf

Dialogue States

The dialogue state consists of all the information given so far (including user-specified constraints on the slots)

User:	I'm looking for a cheaper restaurant
	`inform(price=cheap)`
System:	Sure. What kind - and where?
User:	Thai food, somewhere downtown
	`inform(price=cheap, food=Thai, area=centre)`
System:	The House serves cheap Thai food
User:	Where is it?
	`inform(price=cheap, food=Thai, area=centre); request(address)`
System:	The House is at 106 Regent Street

Slot filling and dialogue act labeling

Given a **user utterance**,

Hello, I'd like an Italian somewhere near the museum

the system needs to identify…

… the **dialogue act label** (e.g. INFORM)

[this is a sequence classification task]

… the **slots** that are being **filled/constrained**

food=Italian, near=museum

[this can be framed as a sequence labeling task]

Dialogue Policy

The system needs to decide what **dialogue act (type + slots)** to realize in its next utterance

$$\text{argmax}_{\text{Act}}\ P(\text{Act} \mid \text{State}, \text{History}, \text{Confidence}, \text{Cost})$$

The probability of the next act may depend on the current dialogue state, the dialogue history (the last utterances), the confidence the system has (e.g. in the ASR transcription), or the cost of making a mistake (e.g. when booking a flight)

Systems also need to decide when to **confirm** or **reject** input

- **Explicit** confirmation: "Did you say you want Italian?"
- **Implicit** confirmation:
  "How expensive should this Italian restaurant be?"
- **Rejection**: "Sorry, I didn't understand what you just said"

Surface realization

Given a **dialogue act**

```
recommend(restaurant name="Au Midi",
          neighborhood = "Midtown",
          cuisine = "french")
```

the system needs to generate an **utterance**

But there may be many possible candidates, e.g

Au Midi is in Midtown and serves French Food.

There is a French Restaurant in Midtown called Au Midi

Surface realization

Given a **dialogue act**

```
recommend(restaurant name="Au Midi",
          neighborhood = "Midtown",
          cuisine = "french")
```

the system needs to generate an **utterance**

But there may be many possible candidates, e.g.:

Au Midi is in Midtown and serves French Food.

There is a French Restaurant in Midtown called Au Midi

These utterances contain the **particular slot values** we are looking for.

Surface realization

Our training data won't contain these sentences, but it might contain instances for other restaurants, e.g.

> LeBec Fin is in Center City and serves French Food.
>
> There is a Cuban Restaurant in Rittenhouse Square called Alma de Cuba

We can **delexicalize** the training data as follows:

> NAME is in NEIGHBORHOOD and serves CUISINE Food.
>
> There is a CUISINE Restaurant in NEIGHBORHOOD called NAME

Now a seq2seq model can be trained on this data, and the slots can be filled in via rules.

Grounded ("situated") dialogue

"Grounding" may also mean that utterances are mapped to/interpreted in a world

- human-robot communication: physical world
- computer games: simulated world
- talking about images/videos: world=images/videos

Increasingly important for communication with smart devices, (self-driving) cars, etc.