

---

# LECTURE 26: DIALOGUE

Mehmet Can Yavuz, PhD.

Adapted from Julia Hockenmaier, NLP S2023 - course material  
<https://courses.grainger.illinois.edu/cs447/sp2023/>



---

# 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: <i>can you tell me how much it is to tokyo?</i> S: You want to know the cost of a call? U: <i>yes that's right</i> S: Please hold on for rate information.
--

S: How may I help you? U: <i>what's the area code for chicago?</i> S: You want an area code. U: <i>(silence)</i> 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 ?

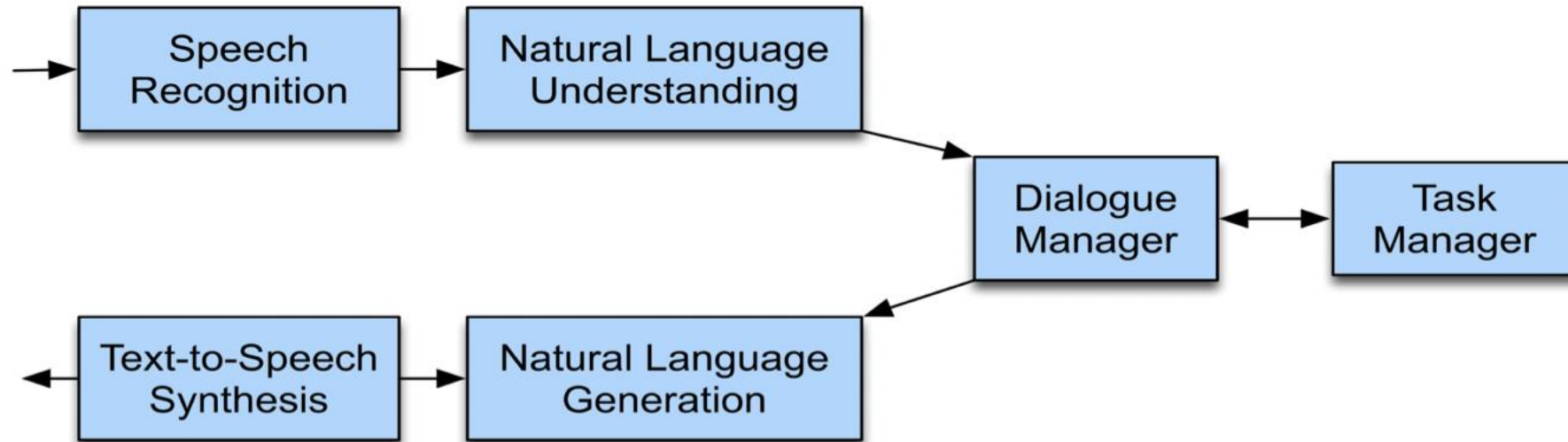
Client: That's fine.

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

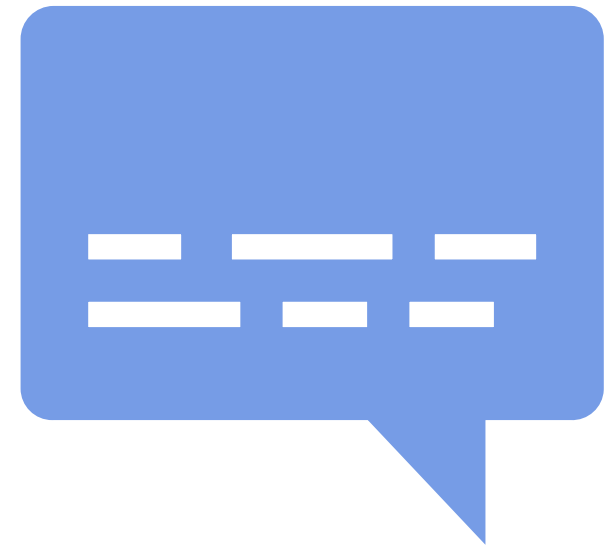




---


# 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

---

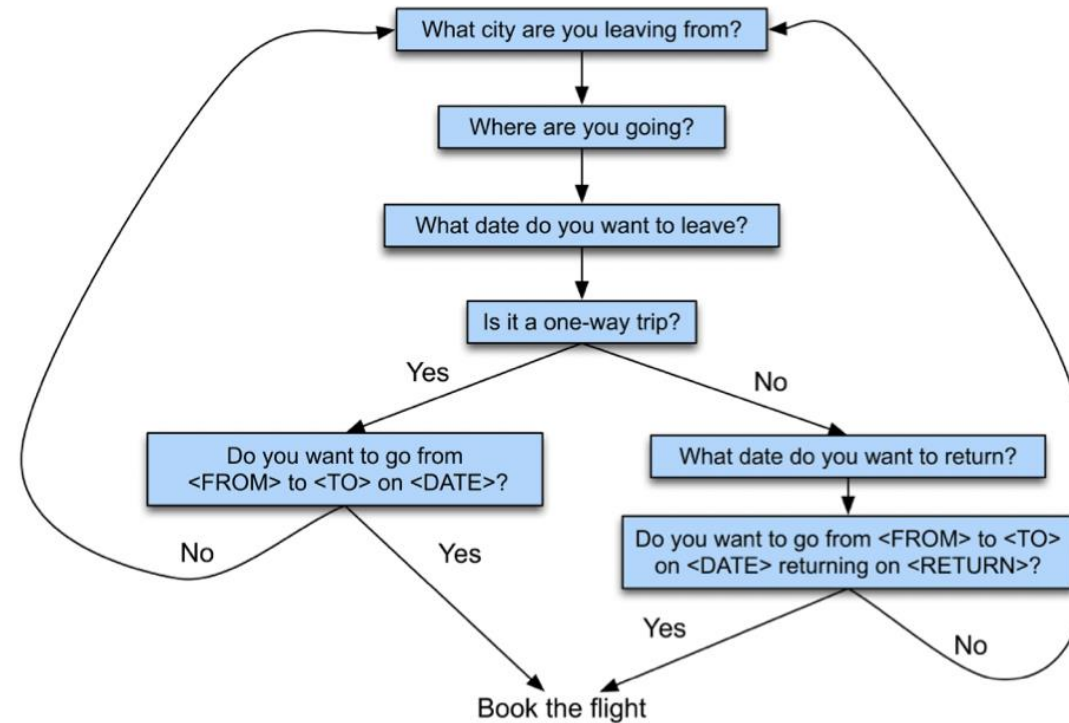
# THE FRAME

- 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 system 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:

[illegible]

- 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...

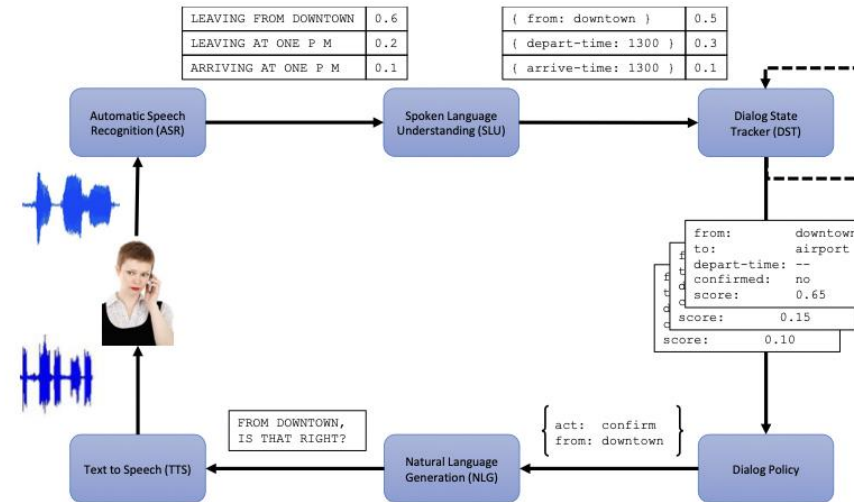
- ...Decide when the user has asked a question,
- made a proposal, rejected a suggestion
- ...Ground a user's utterance, ask clarification questions,
- make new suggestions

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( $a = x, b = y, \dots$ )	✓	✓	Open a dialogue and give info $a = x, b = y, \dots$
INFORM( $a = x, b = y, \dots$ )	✓	✓	Give info $a = x, b = y, \dots$
REQUEST( $a, b = x, \dots$ )	✓	✓	Request value for a given $b = x, \dots$
REQALTS( $a = x, \dots$ )	✓	✓	Request alternative with $a = x, \dots$
CONFIRM( $a = x, b = y, \dots$ )	✓	✓	Explicitly confirm $a = x, b = y, \dots$
CONFREQ( $a = x, \dots, d$ )	✓	✓	Implicitly confirm $a = x, \dots$ and request value of $d$
SELECT( $a = x, a = y$ )	✓	✓	Implicitly confirm $a = x, \dots$ and request value of $d$
AFFIRM( $a = x, b = y, \dots$ )	✓	✓	Affirm and give further info $a = x, b = y, \dots$
NEGATE( $a = x$ )	✓	✓	Negate and give corrected value $a = x$
DENY( $a = x$ )	✓	✓	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

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



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.