Dialogue Systems and Chatbots

Natalie Parde, Ph.D.

Department of Computer Science

University of Illinois at Chicago

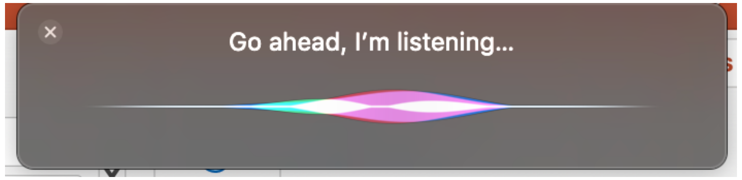
CS 421: Natural Language Processing

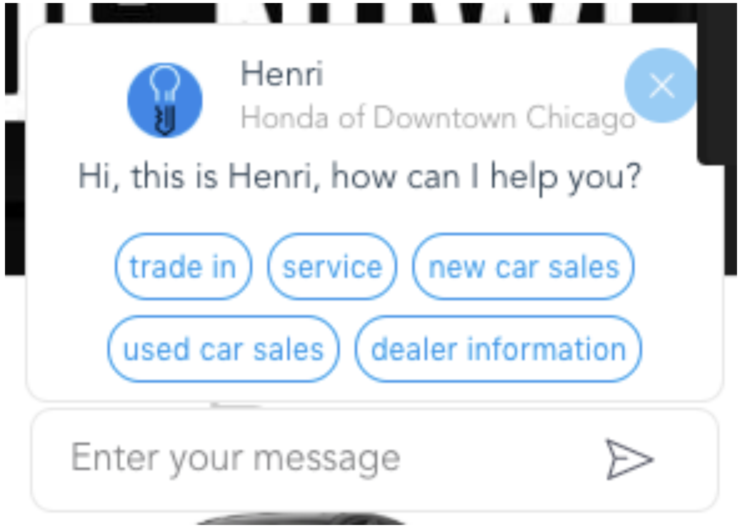
Fall 2019

Many slides adapted from Jurafsky and Martin (https://web.stanford.edu/~jurafsky/slp3/).

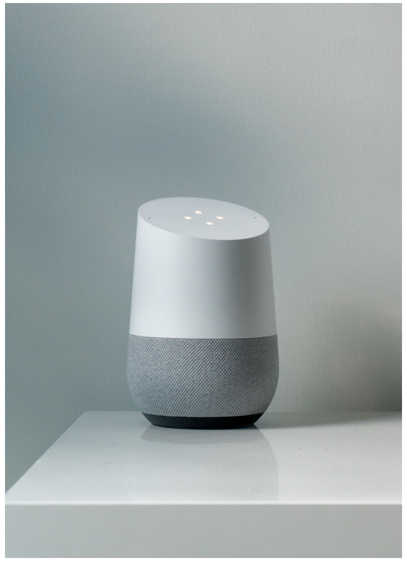
What are dialogue systems?

- **Programs capable of communicating with users** in natural language via text, speech, or both
- Often referred to as **conversational agents**


Go ahead, I'm listening...


Henri
Honda of Downtown Chicago

Hi, this is Henri, how can I help you?



Dialogue systems are everywhere!

Types of Dialogue Systems

Task-Oriented	Chatbots
• Designed to leverage conversational interactions to help users complete tasks	• Designed to carry out extended, unstructured conversations (similar to human chats)

Many dialogue systems contain elements of both of these types!

- Task-oriented agents may seem more natural if they also contain a chatbot component

U: Hey

A: Hi, **how are you**?

U: I'm doing good, how are you?

A: **I'm doing good as well**. **Would you like me to help you reserve a room for your meeting?**

Designing high-quality conversational agents requires an understanding of how human conversation works.

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

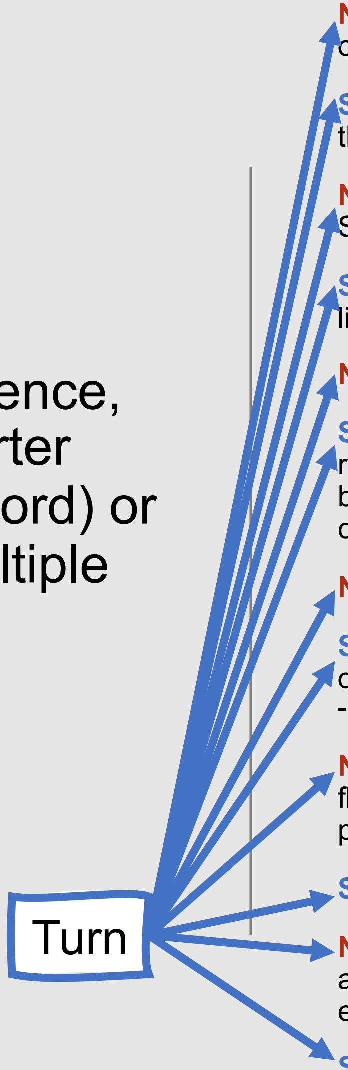
Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Turns:** Individual contributions to the dialogue
  - Typically a sentence, but may be shorter (e.g., a single word) or longer (e.g., multiple sentences)



Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

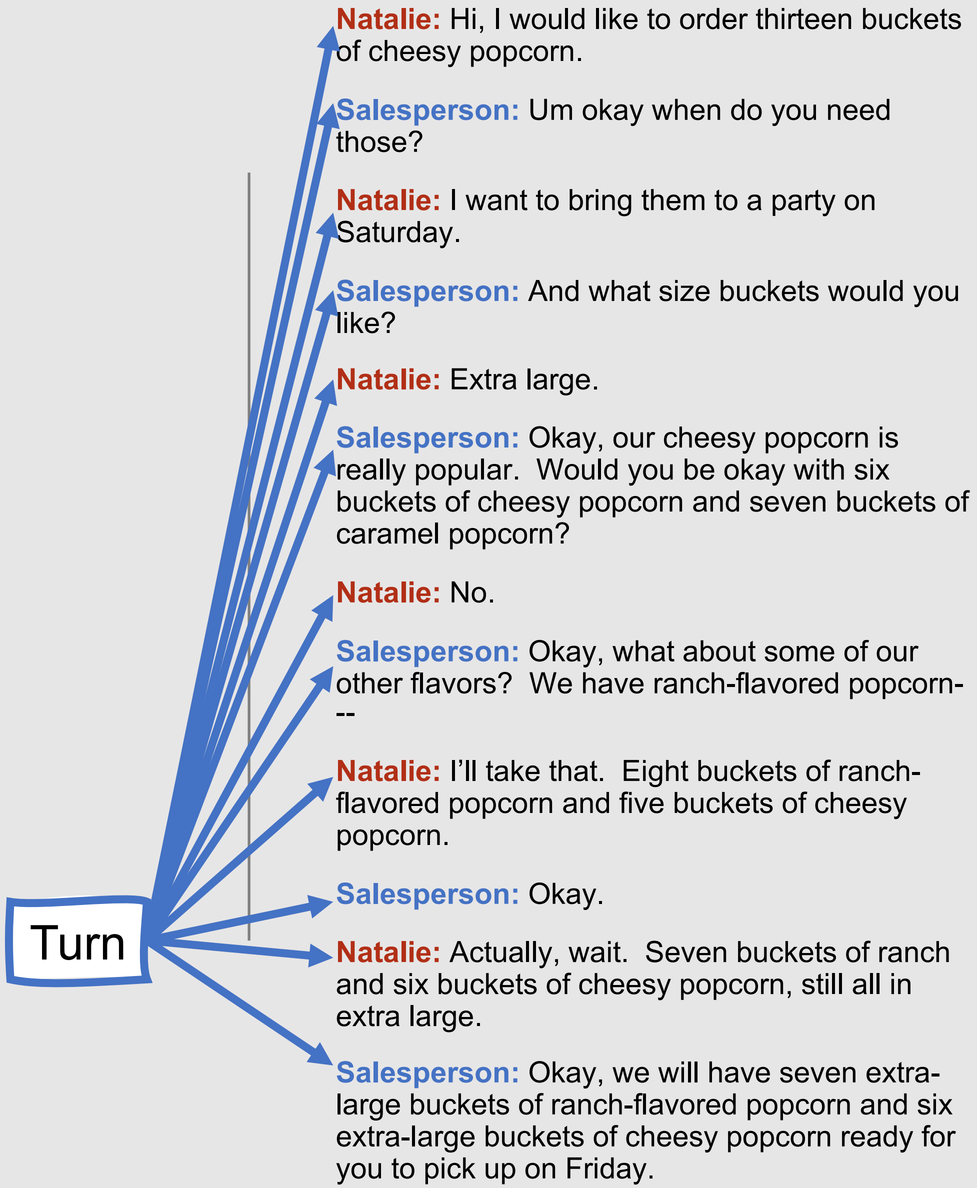
Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- Understanding turn structure is very important for spoken dialogue systems!
- Systems must know:
  - When to stop talking
    - Dealing with interruptions
  - When to start talking
    - Detecting when the human user has finished speaking
- Detecting when a user has finished speaking is called **endpoint detection**
  - Challenging due to noise and speech pauses



Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Speech Acts:** Types of actions performed by the speaker
  - Also referred to as **dialogue acts**
- Major dialogue act groups:
  - **Constatives**
  - **Directives**
  - **Commissives**
  - **Acknowledgments**

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

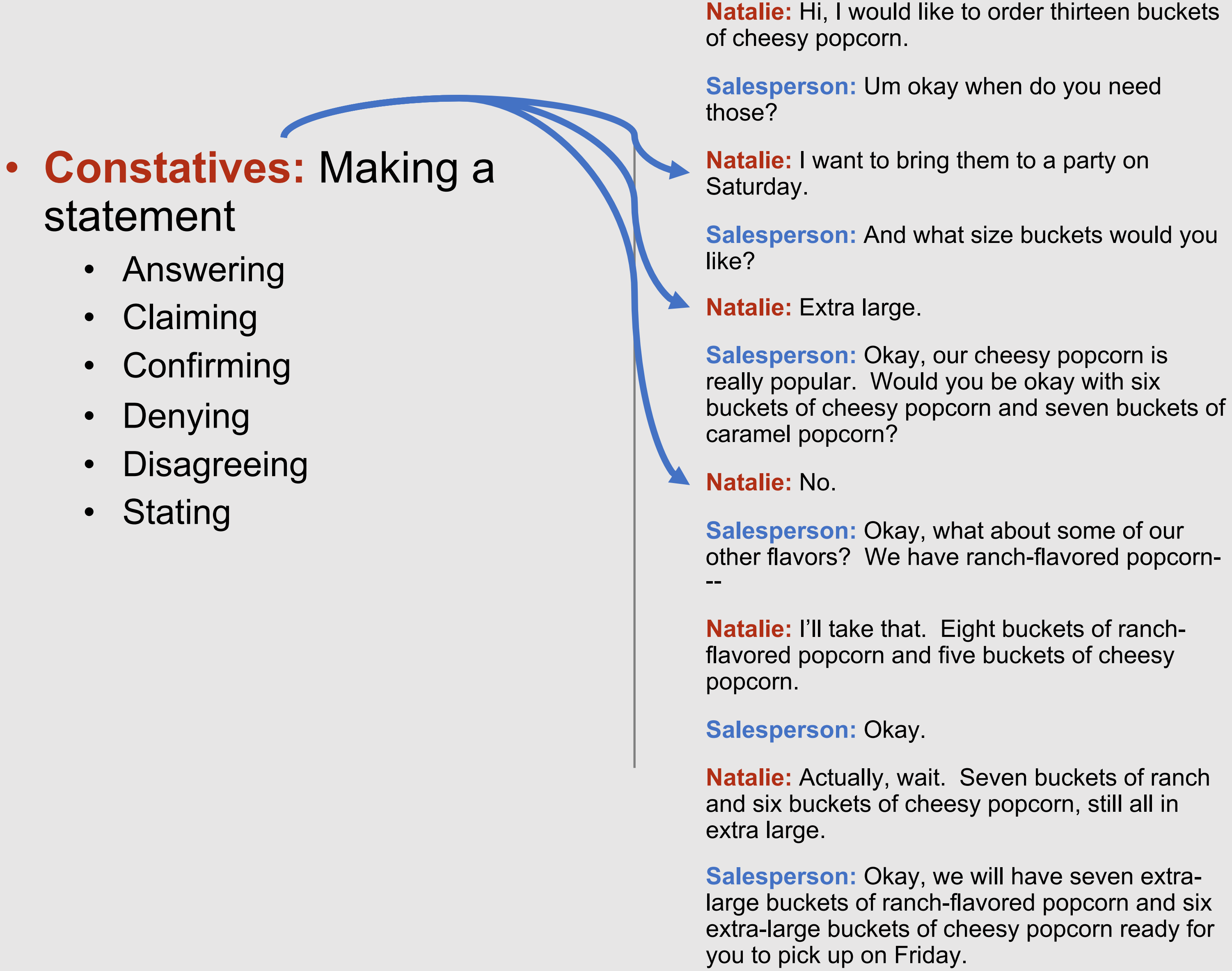
Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation



- **Constatives:** Making a statement
  - Answering
  - Claiming
  - Confirming
  - Denying
  - Disagreeing
  - Stating

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

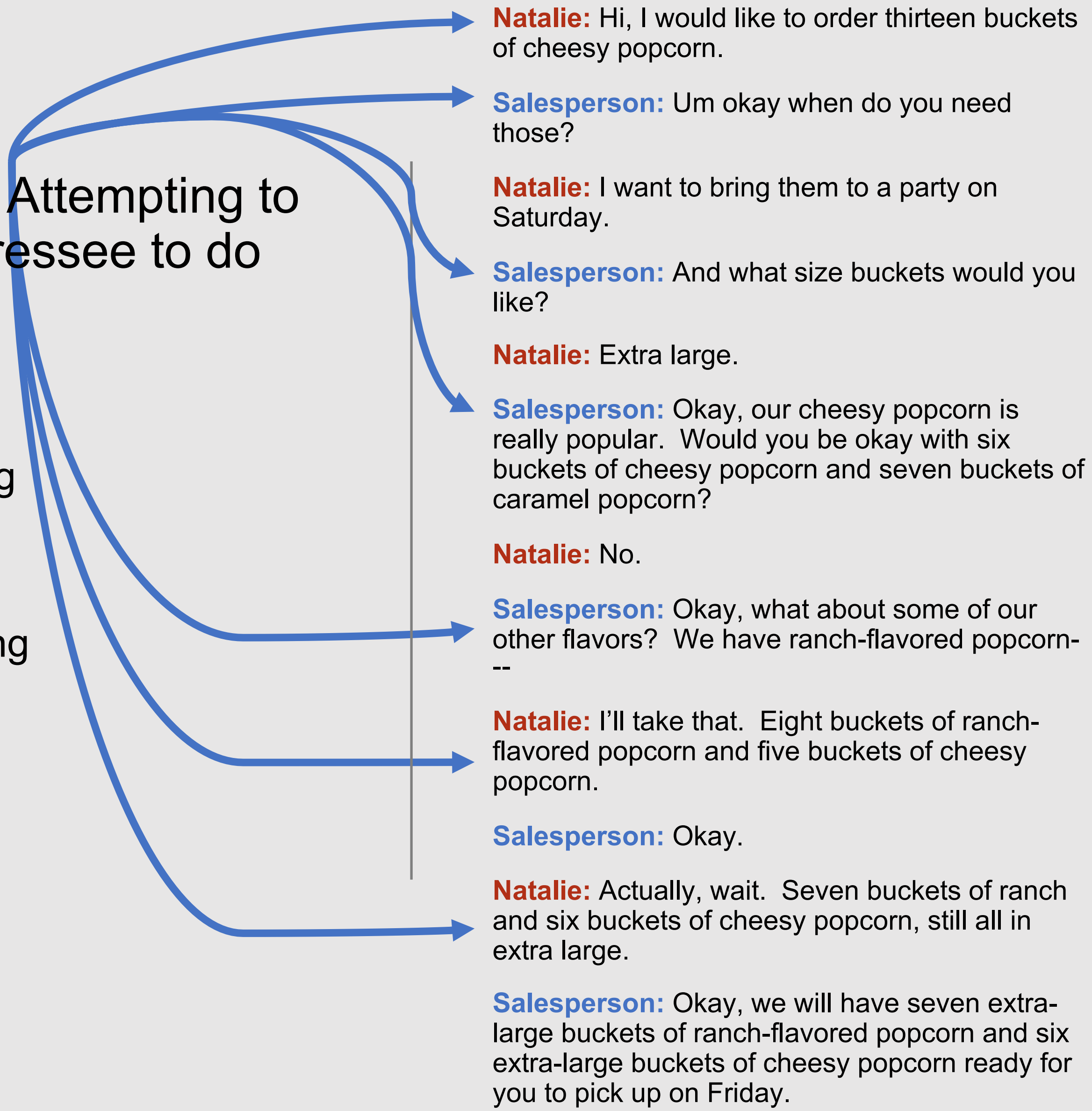
Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Directives:** Attempting to get the addressee to do something
  - Advising
  - Asking
  - Forbidding
  - Inviting
  - Ordering
  - Requesting



Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Commissives:** Committing the speaker to a future action
  - Promising
  - Planning
  - Vowing
  - Betting
  - Opposing

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Acknowlegements:** Expressing the speaker's attitude regarding some social action
  - Apologizing
  - Greeting
  - Thanking
  - Accepting an acknowledgement

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

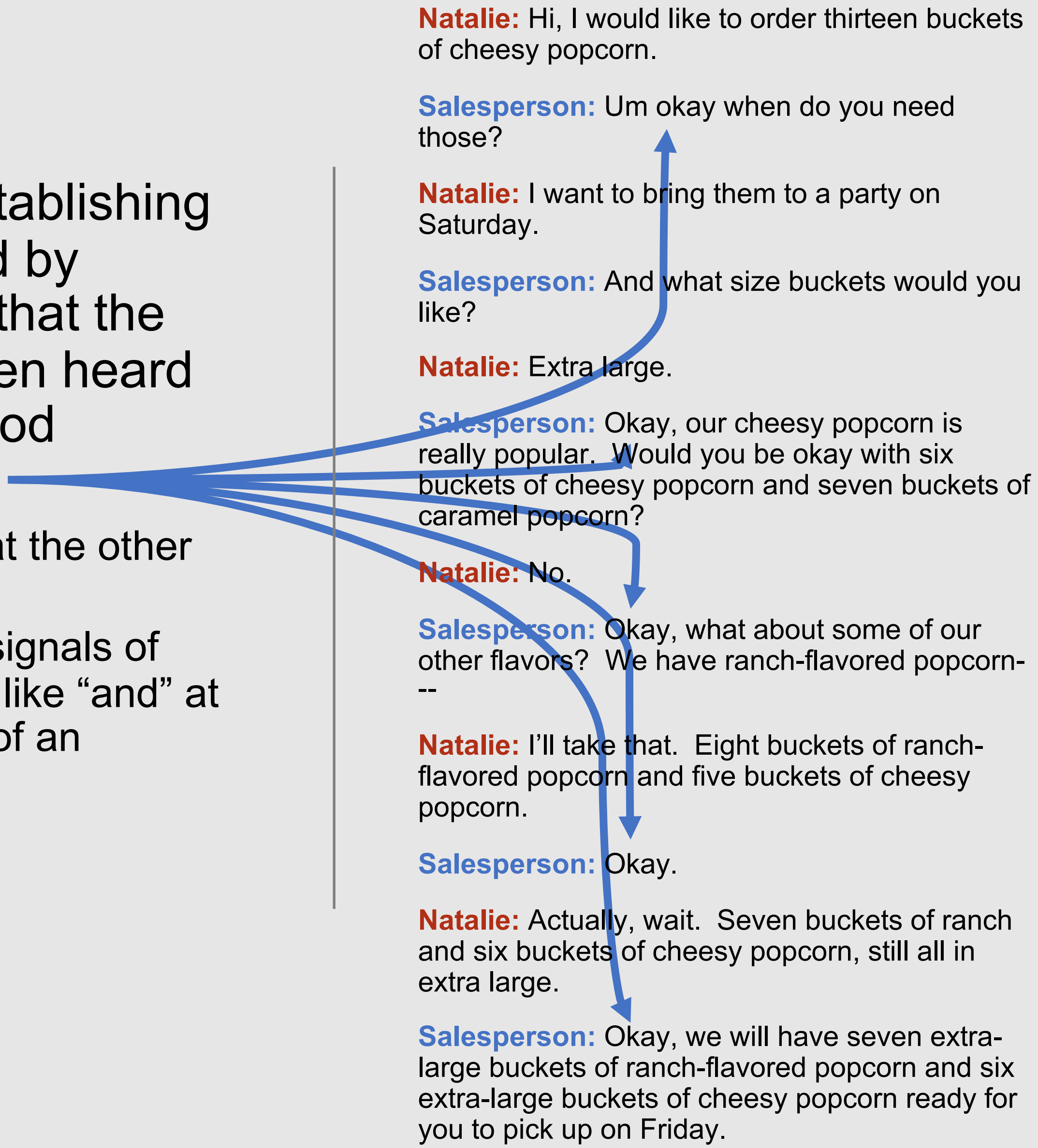
Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Grounding:** Establishing common ground by acknowledging that the speaker has been heard and/or understood
  - Saying "okay"
  - Repeating what the other speaker said
  - Using implicit signals of understanding like "and" at the beginning of an utterance



Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Grounding:** Establishing common ground by acknowledging that the speaker has been heard and/or understood
  - Saying "okay"
  - Repeating what the other speaker said
  - Using implicit signals of understanding like "and" at the beginning of an utterance

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Grounding:** Establishing common ground by acknowledging that the speaker has been heard and/or understood
  - Saying “okay”
  - Repeating what the other speaker said
  - Using implicit signals of understanding like “and” at the beginning of an utterance

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I’ll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- Conversations have structure
  - Questions set up an expectation for an answer
  - Proposals set up an expectation for an acceptance or rejection
- These dialogue act pairs are called **adjacency pairs**
  - First pair part: Question
  - Second pair part: Answer

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- However, two dialogue acts in an adjacency pair don't always immediately follow one another!
- Adjacency pairs can be separated by **side sequences** or **subdialogues**
  - Interruptions
  - Clarifying questions
  - Corrections
- Some adjacency pairs also have **presequences**
  - Requests may be preceded by questions about a system's capabilities

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

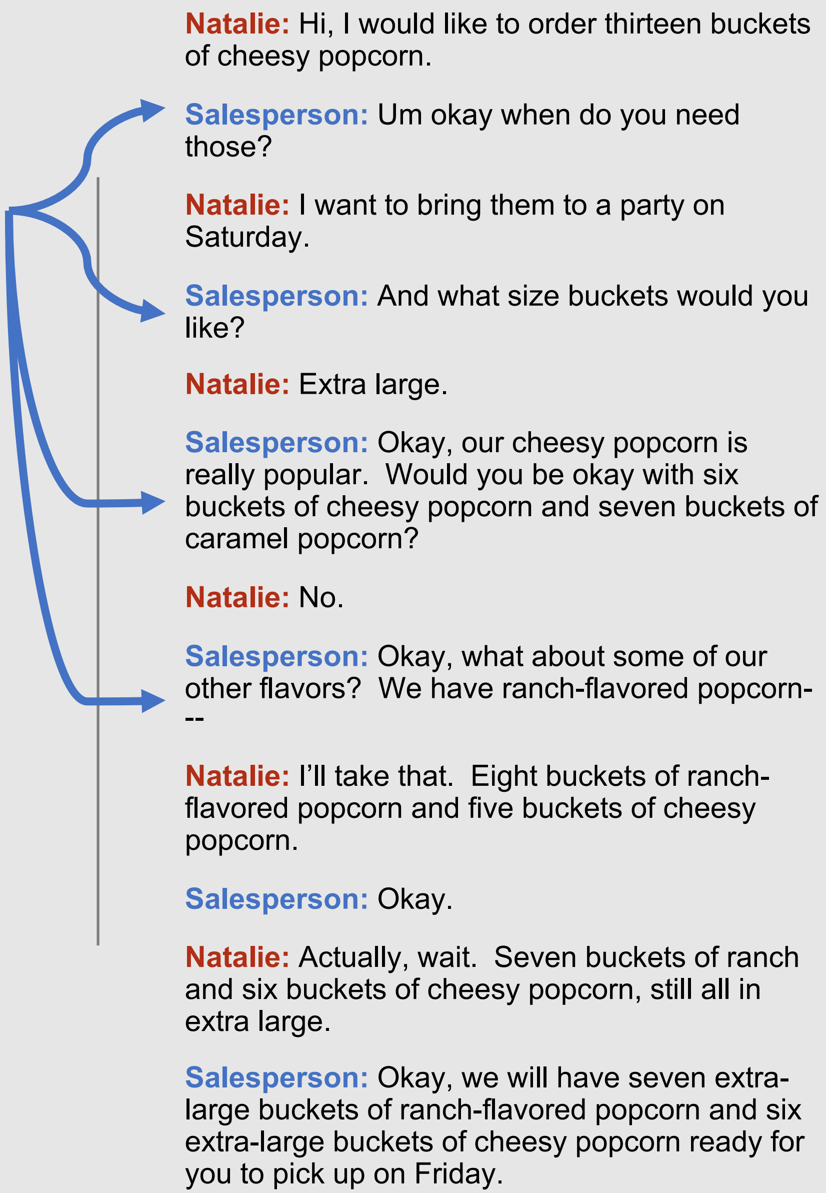
Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Initiative:** Conversational control
- Generally, the speaker asking questions has the **conversational initiative**
- In everyday dialogue, most interactions are **mixed-initiative**
    - Participants sometimes ask questions, and sometimes answer them



Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- Although normal in human-human conversations, mixed-initiative dialogue is very difficult for dialogue systems to achieve!
- In question-answering systems (e.g., “Alexa, what’s the weather like right now?”) the initiative lies entirely with the user
  - Systems such as these are called **user-initiative** systems
- Opposite of user-initiative system: **system-initiative** system
  - Can be very frustrating!

Salesperson: Which variety of caramel popcorn would you like?

Natalie: I don’t want caramel popcorn.

Salesperson: Which variety of caramel popcorn would you like?

Natalie: Can I quit?

Salesperson: Which variety of caramel popcorn would you like?

Natalie: Um, regular.

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn---

Natalie: I’ll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.

Properties of Human Conversation

- **Inference:** Drawing conclusions based on more information than is present in the uttered words
- **Implicature:** The act of implying meaning beyond what is directly communicated

Mentioning the party on Saturday (especially in response to the salesperson's question!) implies that the popcorn will be needed by that time

The salesperson infers that the popcorn should be ready for pickup by Friday

Natalie: Hi, I would like to order thirteen buckets of cheesy popcorn.

Salesperson: Um okay when do you need those?

Natalie: I want to bring them to a party on Saturday.

Salesperson: And what size buckets would you like?

Natalie: Extra large.

Salesperson: Okay, our cheesy popcorn is really popular. Would you be okay with six buckets of cheesy popcorn and seven buckets of caramel popcorn?

Natalie: No.

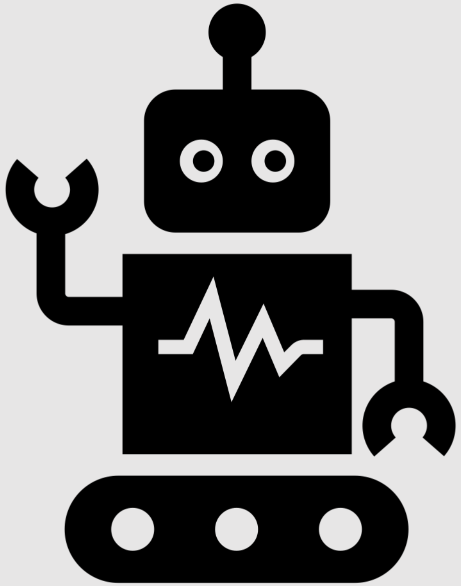
Salesperson: Okay, what about some of our other flavors? We have ranch-flavored popcorn--

Natalie: I'll take that. Eight buckets of ranch-flavored popcorn and five buckets of cheesy popcorn.

Salesperson: Okay.

Natalie: Actually, wait. Seven buckets of ranch and six buckets of cheesy popcorn, still all in extra large.

Salesperson: Okay, we will have seven extra-large buckets of ranch-flavored popcorn and six extra-large buckets of cheesy popcorn ready for you to pick up on Friday.



Chatbots

- Dialogue systems designed to carry out extended conversations with the **goal of mimicking unstructured, informal, human-human chats**
- Can be used for **entertainment**
  - Cleverbot: https://www.cleverbot.com/
- Or more **practical** purposes
  - ELIZA

Chatbot architectures can be divided into two classes.

Rule-based: Interpret and respond to input using a (sometimes extremely complex) set of rules

Corpus-based: Automatically learn to map from a user utterance to a system response based on large datasets of human-human conversations

Rule-based Chatbots

- Revisiting **ELIZA**:
  - Rogerian psychologist (reflect patients' statements right back at them)
  - Takes the stance of knowing almost nothing about the world
    - Common position for chatbots trying to pass the Turing Test …users may assume that the chatbot is pretending not to know anything to fulfill some type of conversational goal

U: Men are all alike.

E: IN WHAT WAY

U: They're always bugging us about something or other.

E: CAN YOU THINK OF A SPECIFIC EXAMPLE

U: Well my boyfriend made me come here.

E: YOUR BOYFRIEND MADE YOU COME HERE

U: He says I'm depressed much of the time.

E: I AM SORRY TO HEAR YOU ARE DEPRESSED

Rule-based Chatbots

- ELIZA works by **matching patterns** and **transforming slots** from those patterns into responses
  - (.* [yY][oO][uU] .* [mM][eE]) → (WHAT MAKES YOU THINK I 3 YOU)
  - You hate me → WHAT MAKES YOU THINK I HATE YOU
- Each pattern used by ELIZA is based on **keyword(s)** that might occur in users' sentences

Rule-based Chatbots

- In ELIZA, keywords are associated with a rank
  - More specific keywords → higher rank
  - More general keywords → lower rank
- Consider two rules:
  - (I .*) → (YOU SAY YOU 2)
  - (EVERYBODY .*) → (WHO IN PARTICULAR ARE YOU THINKING OF)
- Consider an input sentence:
  - “I know everybody made a cooler chatbot.”
- Since “I” is more general than “everybody,” the system will return **WHO IN PARTICULAR ARE YOU THINKING OF** rather than **YOU SAY YOU KNOW EVERYBODY MADE A COOLER CHATBOT**

Rule-based Chatbots

- What if no keywords are matched in an input?
  - ELIZA defaults to a non-committal response
    - PLEASE GO ON
    - THAT’S VERY INTERESTING
    - I SEE

Rule-based Chatbots

- Alternately, if no keywords are matched in an input, ELIZA can access facts from its **memory**
  - Whenever "my" is the highest-ranked keyword, select a transformation rule, apply it to the input, and store the transformed input on a stack
    - My boyfriend made me come here. → DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE
  - Pop the most recent transformed input off the stack if no keywords are matched in an input

Example: ELIZA

Men are all alike.

☺

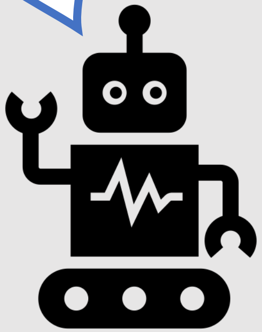
Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

Example: ELIZA

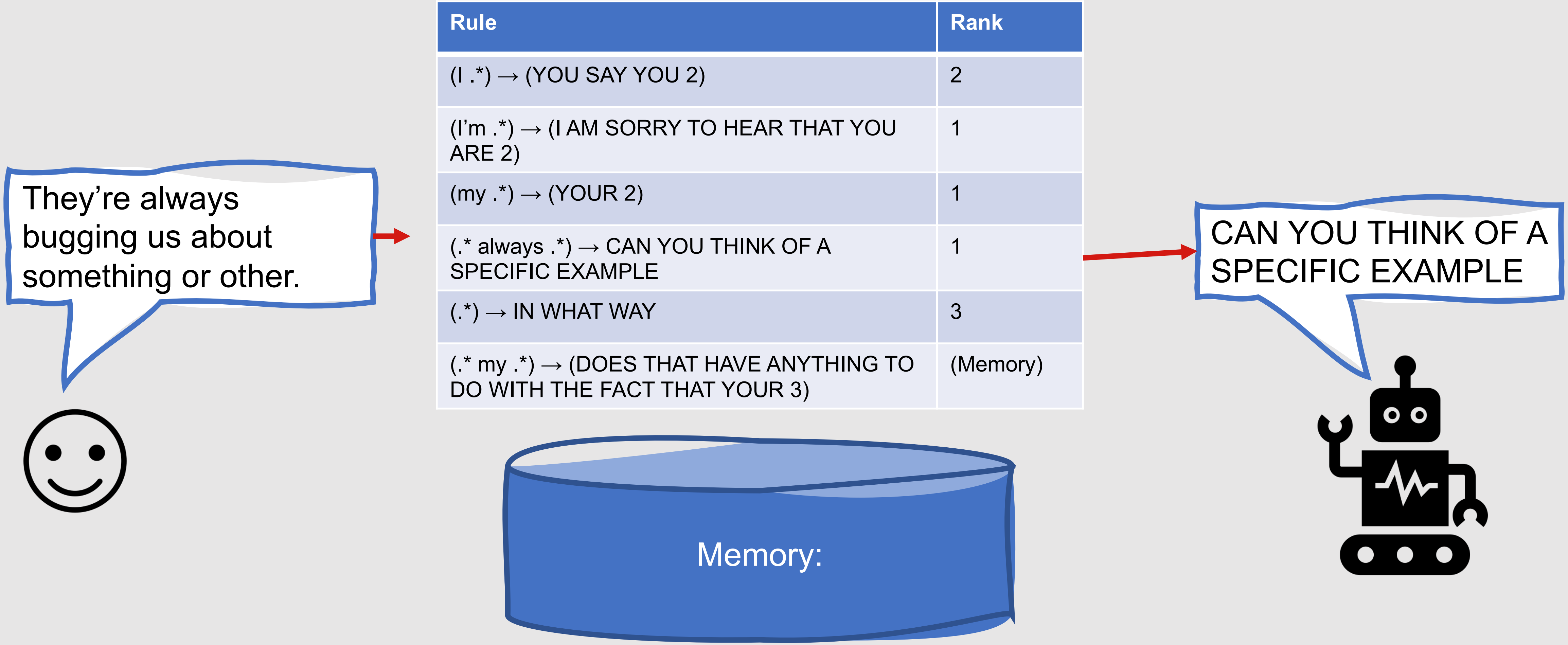
Men are all alike.

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

IN WHAT WAY



Example: ELIZA



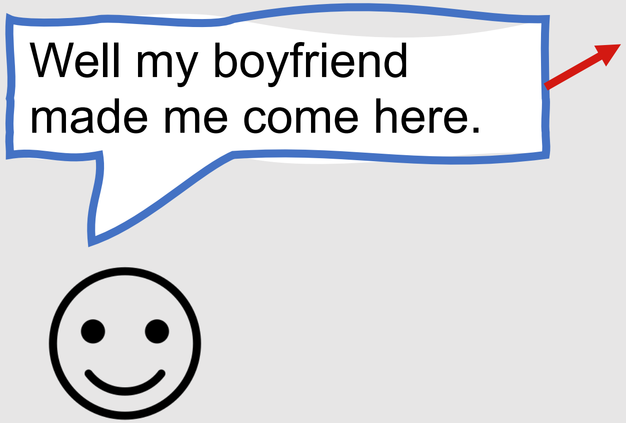
They're always bugging us about something or other.

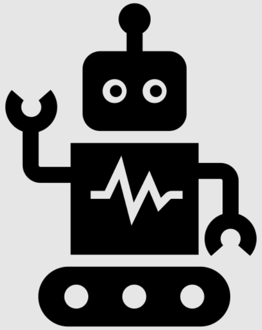
Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

CAN YOU THINK OF A SPECIFIC EXAMPLE

Example: ELIZA

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I’m .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)



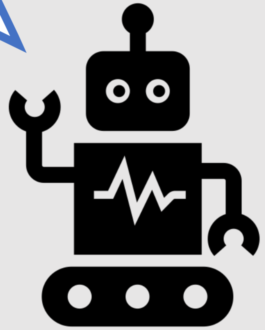


Example: ELIZA

Well my boyfriend made me come here.

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

YOUR BOYFRIEND MADE YOU COME HERE



Memory:

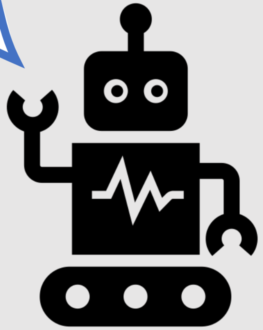
1) DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

Example: ELIZA

He says I’m depressed much of the time.

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I’m .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

I AM SORRY TO HEAR THAT YOU ARE DEPRESSED MUCH OF THE TIME



Memory:
1) DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

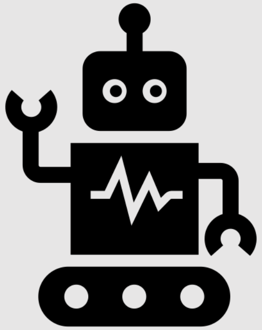
Example: ELIZA

Maybe he is the problem.

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

Memory:

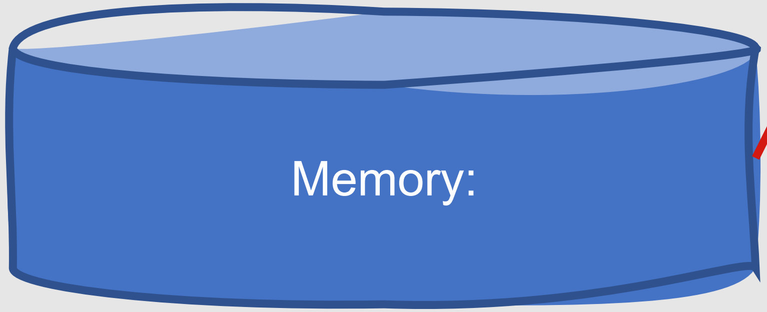
1) DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE



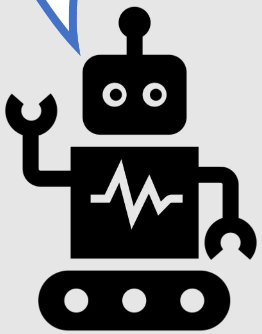
Example: ELIZA

Maybe he is the problem.

Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)



DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE



Formal Algorithm: ELIZA

```
find the word $w$ in the sentence $s$ that has the highest keyword rank
if $w$ exists:
      choose the highest ranked rule $r$ for $w$ that matches $s$
      response ← apply the transformation in $r$ to $s$
      if $w$ == "my":
            future ← apply a transformation from the memory rule
                      list to $s$
            push future onto the memory stack
else: # No keyword applies!
      either:
            response ← apply the transformation for NONE to $s$
      or:
            response ← pop the top response from the memory stack
```

Continued popularity of ELIZA's framework….

- Still used in many applications today!
  - Modern chatbot systems like ALICE are based on updated versions of ELIZA's pattern/action architecture
- PARRY
  - Another clinical psychology chatbot developed several years after ELIZA
  - Included a model of its own mental state, with affect variables for its levels of fear and anger
    - High anger → hostile output
    - High fear → sequence of deluded statements
  - First known system to pass the Turing Test (psychiatrists were unable to distinguish transcripts from PARRY with transcripts from people with real paranoia)

Corpus-based Chatbots

- Do not use hand-built rules
- Instead, learn to map inputs to outputs based on large human-human conversation corpora
- Very data-intensive!
  - Training modern corpus-based chatbots requires hundreds of millions, or even billions, of words

Thanks!
You're welcome.

Doing good, thanks, and you?
I'm doing good as well.

Hello
Hi, how are you?

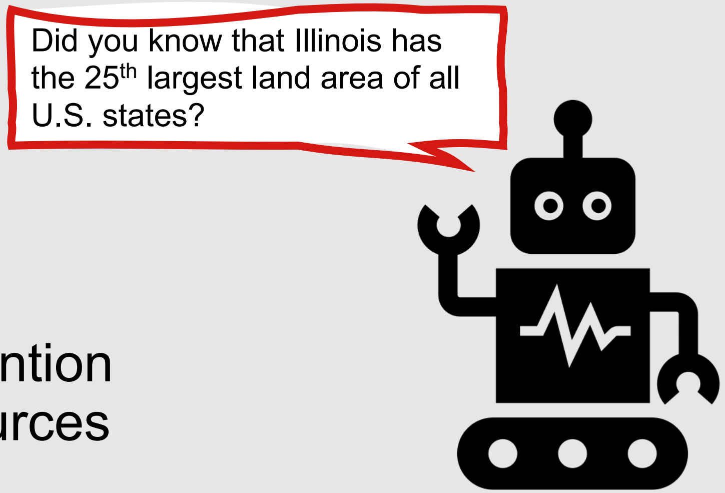
Well, see you later.
Bye!

What kind of corpora are used to train corpus-based chatbots?

- Large spoken conversational corpora
  - Switchboard corpus of American English telephone conversations: https://catalog.ldc.upenn.edu/LDC97S62
- Movie dialogue
- Text from microblogging sites (e.g., Twitter)
- Collections of crowdsourced conversations
  - Topical-Chat: https://github.com/alexa/alexa-prize-topical-chat-dataset

Possible responses can also be extracted from non-dialogue corpora.

- Possible sources:
  - News
  - Online knowledge repositories (e.g., Wikipedia)
- This allows the chatbot to tell stories or mention facts acquired from non-conversational sources


Did you know that Illinois has the 25th largest land area of all U.S. states?

As humans interact with a chatbot, their own utterances can be used as additional training data as well.

- This allows a chatbot's quality to gradually improve over time
- Some privacy concerns can emerge when using this strategy (it's crucial to remove **personally identifiable information**!)

Corpus-based Chatbots

- Two main architectures:
  - **Information retrieval**
  - **Machine learned sequence transduction**
- Most corpus-based chatbots do (surprisingly!) very little modeling of conversational context
- The focus?
  - Generate a single response turn that is appropriate given the user’s immediately previous utterance(s)

Corpus-based Chatbots

- Since they tend to rely only on very recent utterances, many corpus-based chatbots are viewed more as **response generation** systems
- This makes them similar to **question answering systems**:
  - Focus on single responses
  - Ignore context or larger conversational goals

Information Retrieval-based Chatbots

- Respond to a user’s turn by **repeating some appropriate turn from a corpus** of natural human conversational text
- Any information retrieval algorithm can be used to **choose the appropriate response**
- Two simple methods:
    - Return the response to the most similar turn
    - Return the most similar turn

How can we return the response to the most similar turn?

- Look for a turn that resembles the user's turn, and return the human response to that turn
- More formally, given:
  - A user query, *q*
  - A conversational corpus, *C*
- Find the turn *t* in *C* that is most similar to *q* (e.g., using cosine similarity) and return the human response to *t*
  - $r = \text{response}(\underset{t \in C}{\operatorname{argmax}} \frac{q^T t}{\|q\| \|t\|})$

How can we return the most similar turn?

- Directly match the user’s query, *q*, with turns from *C*, since a good response will often share words or semantic patterns with the prior turn
- More formally:
  - $r = \underset{t \in C}{\operatorname{argmax}} \frac{q^T t}{\|q\| \|t\|}$

Which of these methods works better?

Depends on the application

More often, returning the most similar turn seems to work better

Slightly less noise from the alternate method of selecting the response to the most similar turn

Various techniques can be used to improve performance with IR-based chatbots.

- Possible additional features:
  - **Entire conversation** with the user so far
    - Particularly useful when dealing with short user queries, e.g., “yes”
  - **User-specific** information
  - **Sentiment**
  - Information from **external knowledge sources**

Encoder-Decoder Chatbots

- **Machine learned sequence transduction:** System learns from a corpus to **transduce a question to an answer**
  - Machine learning version of ELIZA
- Intuition borrowed from **phrase-based machine translation**
  - Learn to convert one phrase of text into another
- Key difference?
  - In phrase-based machine translation, words or phrases in the source and target sentences tend to align well with one another
  - In response generation, a **user's input might share no words or phrases with a coherent, relevant response**

How does a chatbot learn to perform this transduction?

Encoder-decoder models

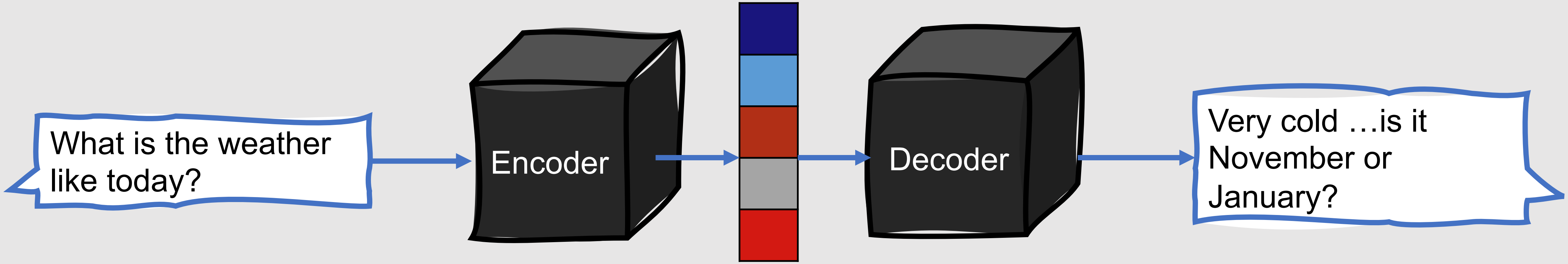
- Accept sequential information as input, and return different sequential information as output

Also recently used in:

- Machine translation
- Question answering
- Summarization

How do encoder-decoder models work?

- In NLP applications, encoders and decoders are often some type of **RNN**
- Encoders take sequential input and generate an **encoded representation** of it
  - This representation is comprised of the outputs from the last hidden state of the encoder network …it is undecipherable to casual observers!
- Decoders take this representation as input and generate a sequential (interpretable) output



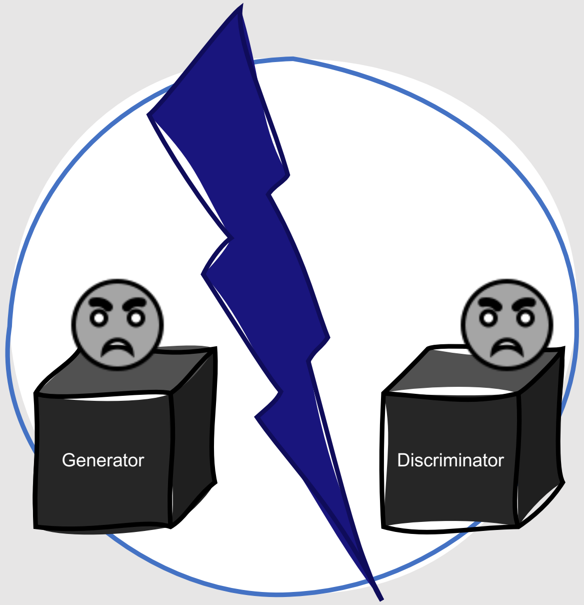
What is the weather like today?

Very cold …is it November or January?

Encoder-Decoder Chatbots

- Basic encoder-decoder models tend to produce repetitive (and therefore boring) responses that don’t encourage further conversation
  - “I’m okay”
  - “I don’t know”
- To avoid this, it is important to incentivize response diversity
  - Mutual information objective function
  - Beam search

Encoder-Decoder Chatbots



- Other challenges?
  - Inability to model prior context
    - Can be solved by using a **hierarchical model** that summarizes information over multiple turns
  - Often poor multi-turn coherence
    - Can be addressed to some extent using **reinforcement learning** or **adversarial networks** to learn to choose responses that make the overall conversation more natural

How are chatbots evaluated?

Currently best evaluation technique?

- Collect human ratings!

Automated metrics correlate poorly with human judgements

- Typically measure slot-filling accuracy or word overlap with a gold standard
- These metrics work best when there are few possible responses and the responses lexically overlap with one another

Task-based Dialogue Systems

- Dialogue system has a specific **goal** (generally, helping a user solve some task)
  - Making a reservation
  - Purchasing a product
- Task-based dialogue systems are **frame-based**
  - Assume a set of user **intentions**, each of which consist of a collection of **slots** that can be filled by possible **values**
  - Set of **frames** is sometimes called a **domain ontology**

Frame-based Dialogue Systems

- Simple (but very long-lived!) frame-based dialogue system: GUS
  - Genial Understander System
  - Originally used for travel planning (Bobrow et al., 1977)

GUS: Hello. My name is GUS. I can help you plan a 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: When 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.

Components of Frame-based Dialogue Systems

Slots: Specify what the system needs to know

- **Slot fillers** are constrained to values of specific semantic types
  - In the travel domain, these could be *cities*, *dates*, *airlines*, or *times*

These types may have hierarchical structures

- In GUS, *date* is a frame itself, with slots for *month*, *year*, *day*, and *weekday*

Components of Frame-based Dialogue Systems

Slot	Type	Question Template
ORIGIN CITY	city	“From what city are you leaving?”
DESTINATION CITY	city	“Where are you going?”
DEPARTURE TIME	time	“When would you like to leave?”
DEPARTURE DATE	date	”What date would you like to leave?”
ARRIVAL TIME	time	“When do you want to arrive?”
ARRIVAL DATE	date	“What day would you like to arrive?”

Control Structure for Frame-based Dialogue

The control architecture for frame-based dialogue systems is also designed around the frame

- Goal:
  1. Fill the slots in the frame with the fillers the user intends
  2. Perform the relevant action for the user

The system achieves its goal by asking questions of a user

- Typically these questions are constructed using pre-specified question templates associated with each slot of each frame

Control Structure for Frame-based Dialogue

- The system continues questioning the user until it is able to fill all slots needed to perform the desired task
- GUS attaches **condition-action rules** to slots to reduce monotony
  - If a user has specified a flight destination city, it may automatically fill the hotel destination slot with that value as well

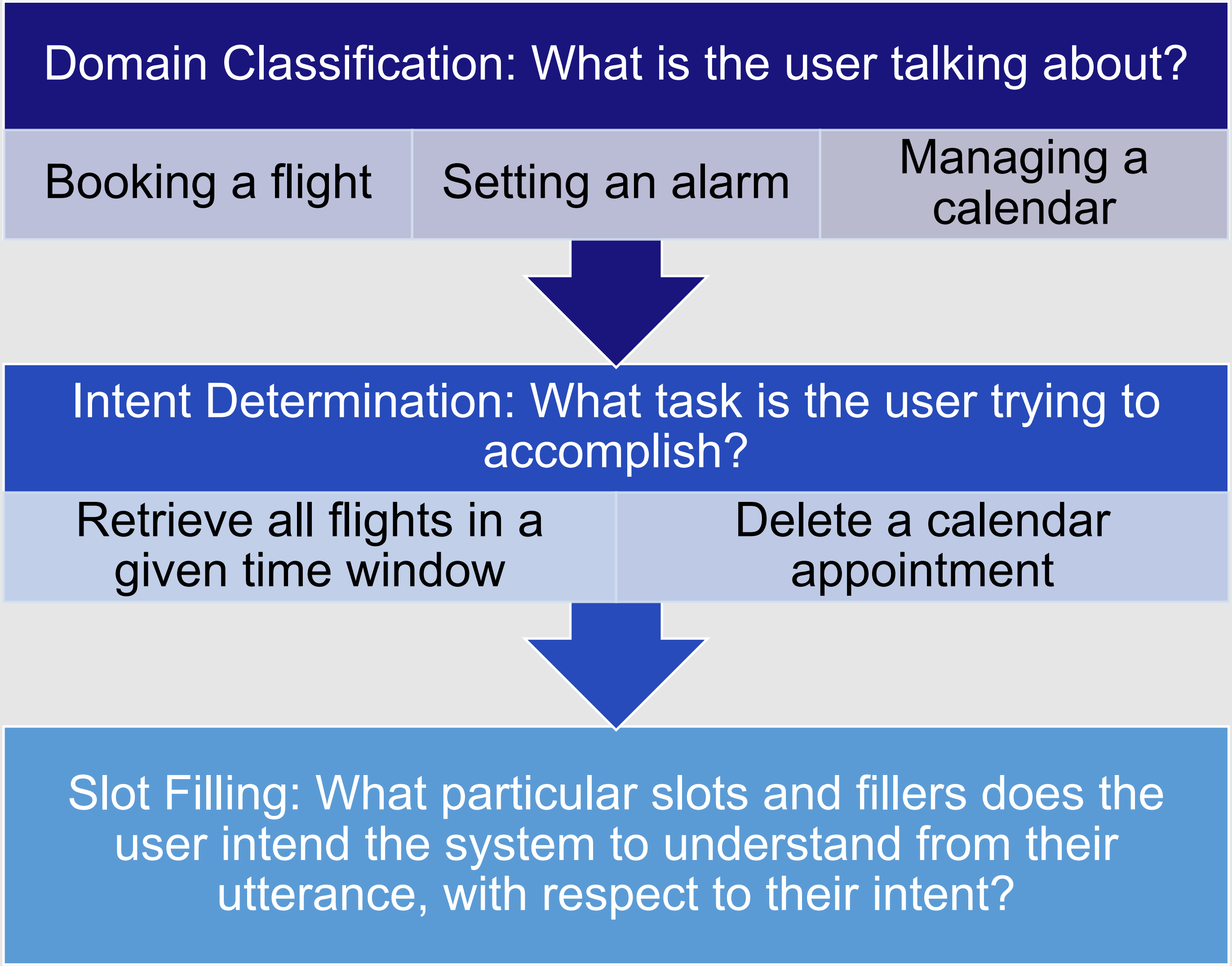
Control Structure for Frame-based Dialogue

- Many domains require multiple frames!
- Dialogue systems must be able to **disambiguate** which slot of which frame a given input is supposed to fill, and then switch dialogue control to that frame
- This can be done using **production rules**
  - Different types of inputs and recent dialogue history match different frames
  - Control is switched to the matched frame
- Once the system has enough information, it performs the desired task (e.g., querying a database of flights) and returns the result to the user

Natural Language Understanding in Frame-based Dialogue Systems

- In a frame-based dialogue system, natural language understanding is necessary for performing three tasks:
  - **Domain classification**
  - **Intent determination**
  - **Slot filling**

Natural Language Understanding in Frame-based Dialogue Systems



Domain Classification: What is the user talking about?

Booking a flight	Setting an alarm	Managing a calendar

Intent Determination: What task is the user trying to accomplish?

Retrieve all flights in a given time window	Delete a calendar appointment

Slot Filling: What particular slots and fillers does the user intend the system to understand from their utterance, with respect to their intent?

Natural Language Understanding for Slot Filling in Frame-based Dialogue Systems

Show me the morning flights from Chicago to Dallas on Thursday.

Domain: AIR-TRAVEL
Intent: SHOW FLIGHTS
Origin-City: Chicago
Origin-Date: Thursday
Origin-Time: morning
Destination-City: Dallas

Wake me tomorrow at 6

Domain: ALARM-CLOCK
Intent: SET-ALARM
Time: 2019-11-13 0600

Natural Language Understanding for Slot Filling in Frame-based Dialogue Systems

In GUS, and in many commercial applications, slots are filled using handwritten rules

wake me (up)? | set (the|an) alarm | get me up → Intent: SET-ALARM

Rule-based systems often include large quantities (thousands!) of rules structured as semantic grammars

Semantic Grammar: A context-free grammar in which the left-hand side of each rule corresponds to the semantic entities (slot names) being expressed

Semantic grammars can be parsed using any CFG parsing algorithm

Other systems use supervised learning for slot filling

Semantic Grammar

SHOW → show me | i want | can i see

DEPART_TIME_RANGE → (after | around | before) HOUR | morning | afternoon | evening

HOUR → one | two | three | four | … | twelve (AM|PM)

FLIGHTS → (a) flight | flights

AMPM → am | pm

ORIGIN → from CITY

DESTINATION → to CITY

CITY → Chicago | Dallas | Denver | Phoenix

Other Components of Frame-based Dialogue Systems

- **Automated Speech Recognition:** Converts audio input to a string of text
  - May or may not be constrained based on the current dialogue domain and/or intent
- **Natural Language Generation:** Produces the utterances that the system outputs to the user
  - Frame-based systems typically use **template-based generation**
    - What time do you want to leave ORIGIN-CITY?
    - Will you return to ORIGIN-CITY from DESTINATION-CITY?
- **Text to Speech Synthesis:** Converts a string of text to an audio output
  - May be done at runtime or using prerecorded statements or phrases

Advantages and Disadvantages of GUS

Why use a simple, rule-based architecture for task-oriented dialogue systems?

- High precision
- Given a narrow domain and sufficient expertise, can provide sufficient coverage

Why explore other options?

- Handwritten rules or grammars can be expensive and slow to create
- Low recall

Summary: Dialogue Systems and Chatbots (Part 1)

- **Dialogue systems** are programs capable of communicating with users in natural language
- Two types of dialogue systems:
  - **Conversational (chatbot)**
  - **Task-oriented**
- Dialogue systems must understand a variety of communicative functions, including **turn-taking**, **dialogue acts**, **grounding**, **conversational structure**, **initiative**, and **implicature**
- Chatbots can be **rule-based** or **corpus-based**
- Corpus-based chatbots can use **information retrieval** or **sequence transduction** methods
- Task-based dialogue systems are **frame-based**
- Frame-based dialogue systems assume a set of **intentions**, each of which consist of a set of **slots** that can be filled by possible **values**
- Slots can be filled using **rule-based** or **machine learning** approaches

More Sophisticated Frame-based Dialogue Systems

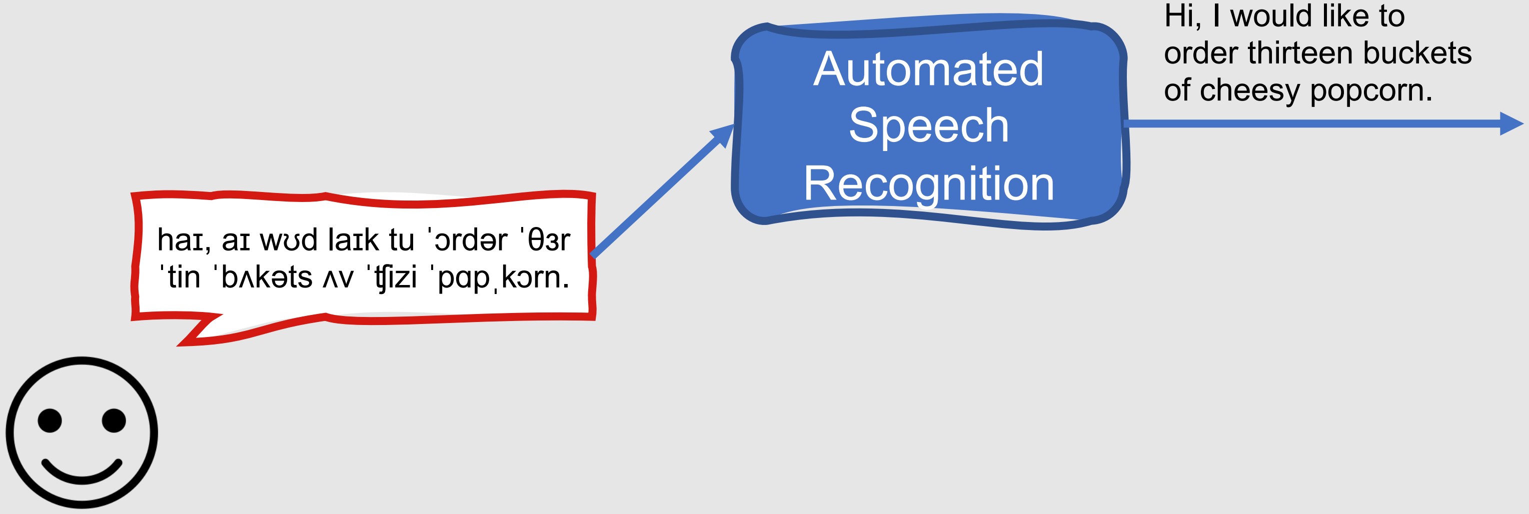
- **Dialogue-State** (or **Belief-State**) **Architecture:** A modular dialogue system architecture comprised of six main components:
  - **Automated Speech Recognition (ASR)**
  - **Natural Language Understanding (NLU)**
  - **Dialogue State Tracker**
  - **Dialogue Policy**
  - **Natural Language Generation (NLG)**
  - **Text to Speech (TTS)**

Dialogue State Architecture

haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

☺

Dialogue State Architecture

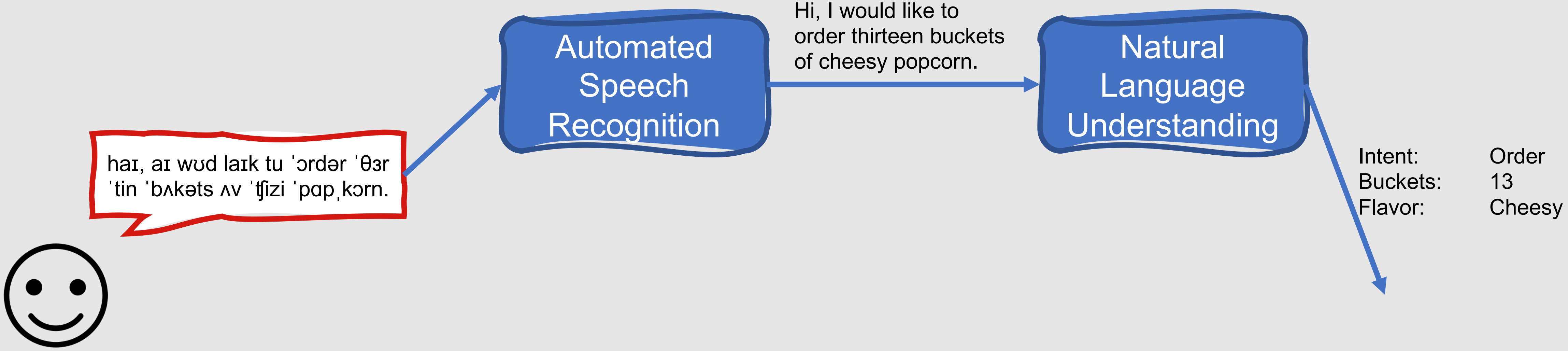


haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Dialogue State Architecture



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

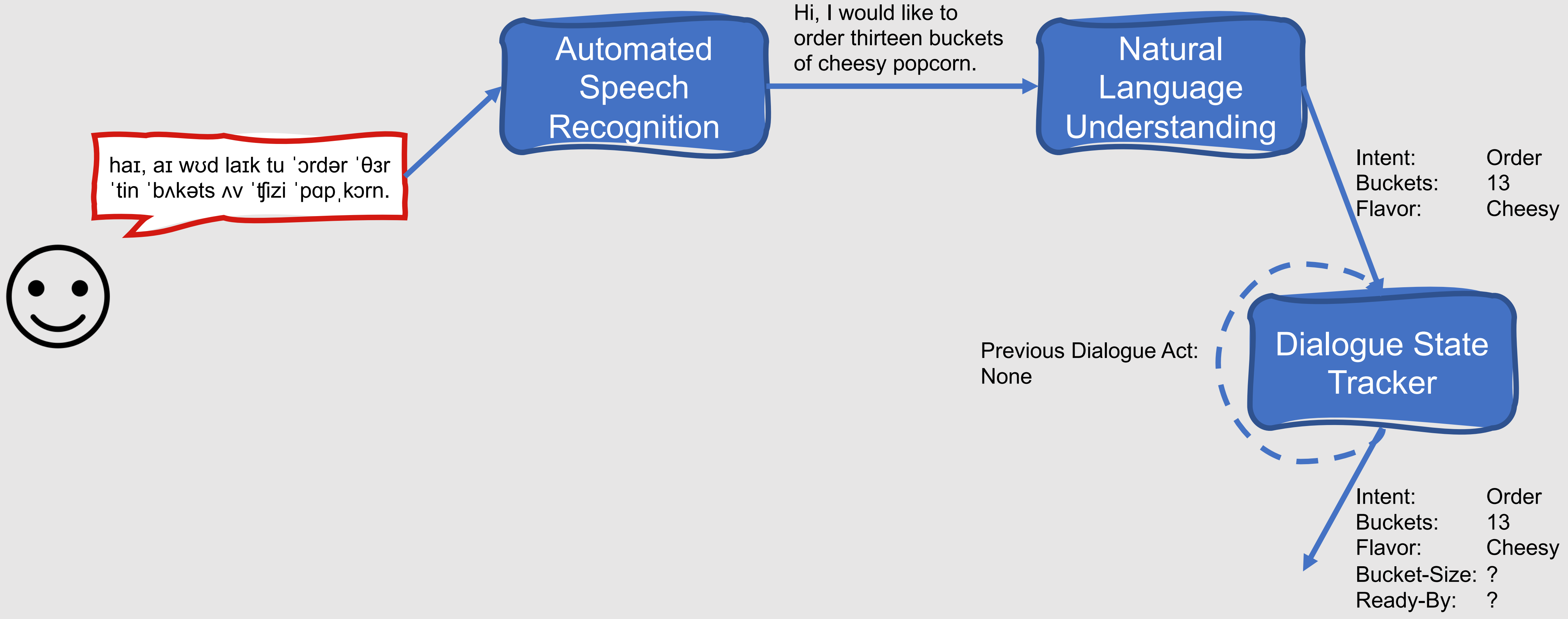
Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Dialogue State Architecture



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

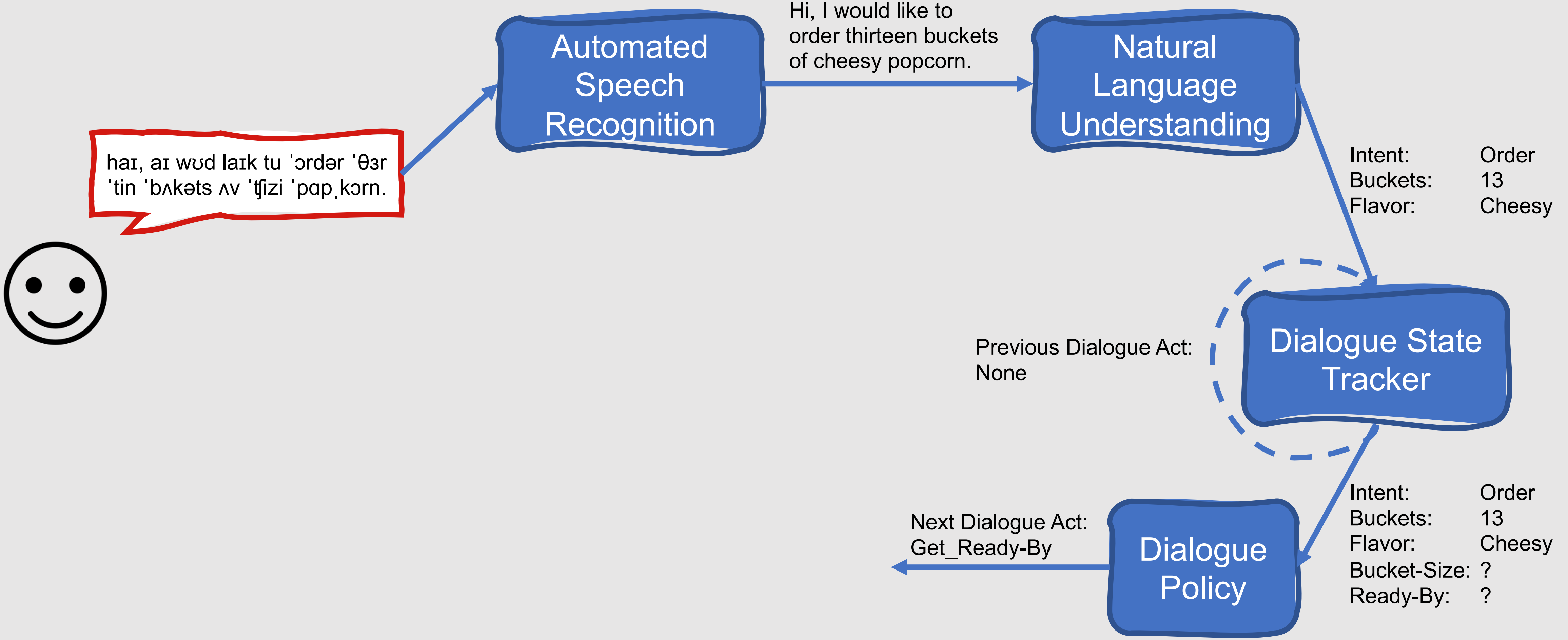
Intent: Order
Buckets: 13
Flavor: Cheesy

Previous Dialogue Act: None

Dialogue State Tracker

Intent: Order
Buckets: 13
Flavor: Cheesy
Bucket-Size: ?
Ready-By: ?

Dialogue State Architecture



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Previous Dialogue Act: None

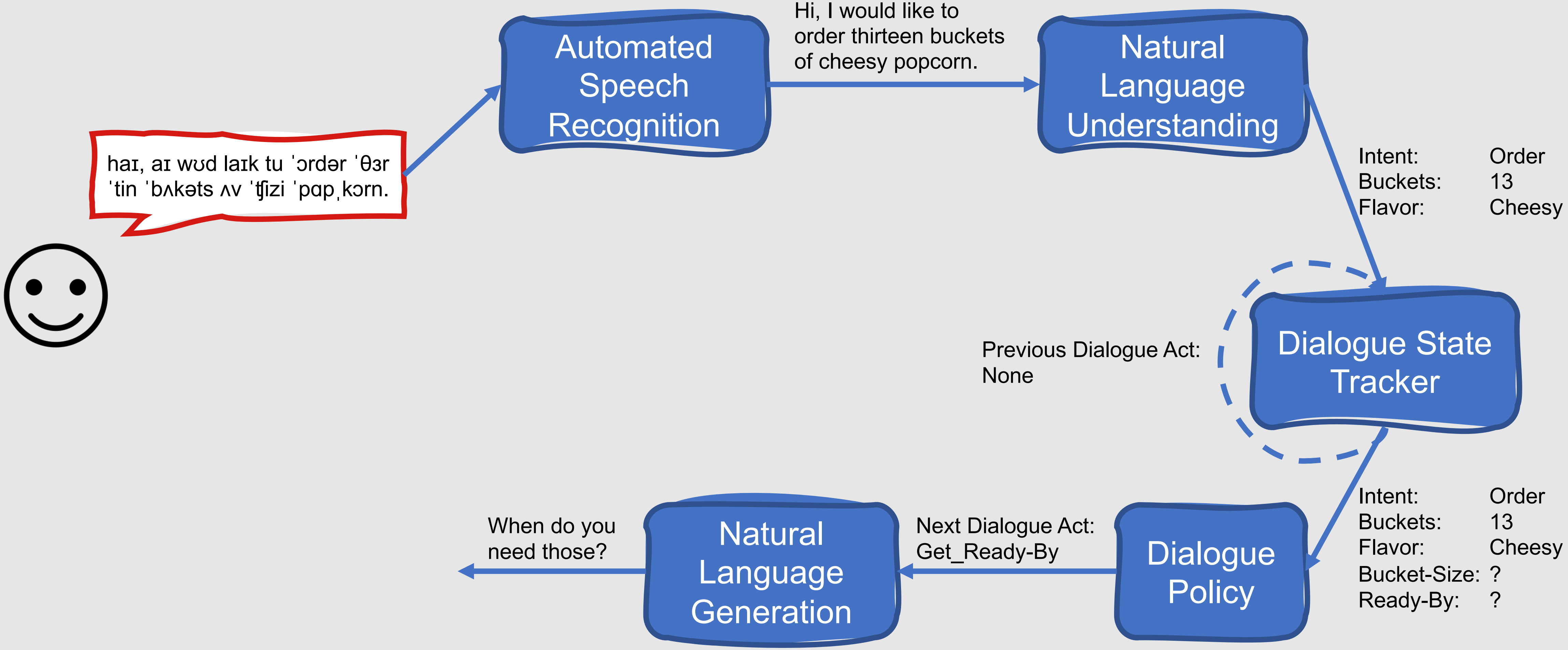
Dialogue State Tracker

Intent: Order
Buckets: 13
Flavor: Cheesy
Bucket-Size: ?
Ready-By: ?

Dialogue Policy

Next Dialogue Act: Get_Ready-By

Dialogue State Architecture



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Dialogue State Tracker

Previous Dialogue Act: None

Intent: Order
Buckets: 13
Flavor: Cheesy
Bucket-Size: ?
Ready-By: ?

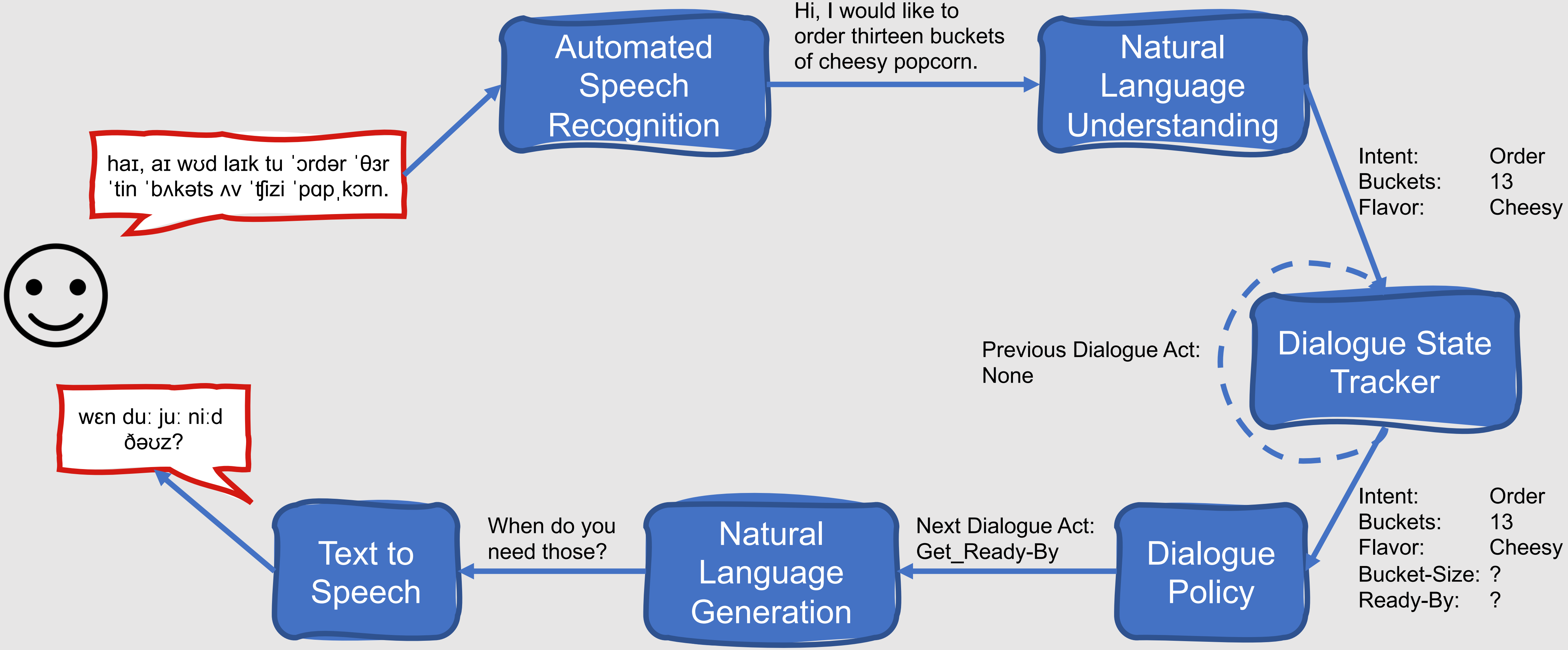
Dialogue Policy

Next Dialogue Act: Get_Ready-By

Natural Language Generation

When do you need those?

Dialogue State Architecture



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent:	Order
Buckets:	13
Flavor:	Cheesy

Dialogue State Tracker

Previous Dialogue Act:
None

Intent:	Order
Buckets:	13
Flavor:	Cheesy
Bucket-Size:	?
Ready-By:	?

Dialogue Policy

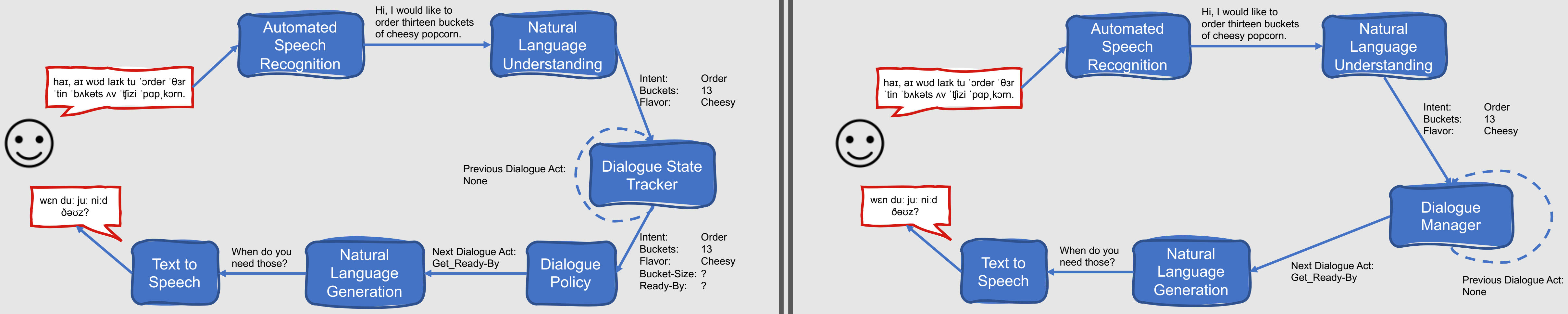
Next Dialogue Act:
Get_Ready-By

Natural Language Generation

When do you need those?

Text to Speech

wɛn duː juː niːd ðəʊz?



haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Previous Dialogue Act: None

Dialogue State Tracker

Intent: Order
Buckets: 13
Flavor: Cheesy
Bucket-Size: ?
Ready-By: ?

Dialogue Policy

Next Dialogue Act: Get_Ready-By

Natural Language Generation

When do you need those?

Text to Speech

wɛn du: ju: ni:d ðəʊz?

haɪ, aɪ wʊd laɪk tu ˈɔrdər ˈθɜr ˈtin ˈbʌkəts ʌv ˈʧizi ˈpɑpˌkɔrn.

Automated Speech Recognition

Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Dialogue Manager

Previous Dialogue Act: None

Next Dialogue Act: Get_Ready-By

Natural Language Generation

When do you need those?

Text to Speech

wɛn du: ju: ni:d ðəʊz?

The dialogue state tracker and dialogue policy are sometimes grouped together as a single dialogue manager.

Automated Speech Recognition

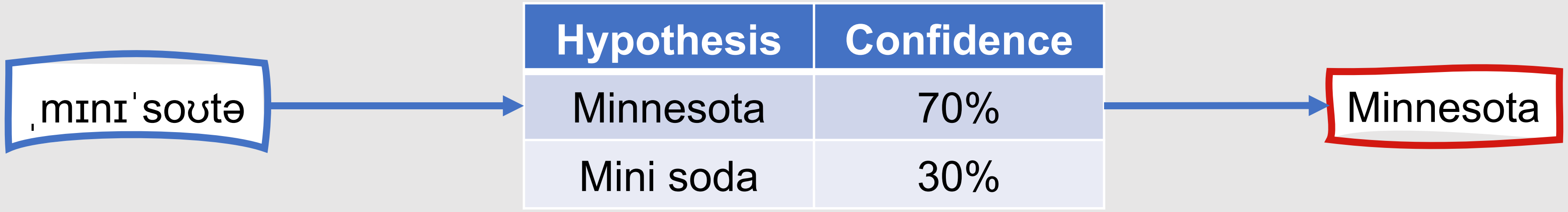
- Input: Audio
- Output: Transcribed string of words
- Can be optimized for domain-dependent dialogue systems by constraining the vocabulary to a fixed, smaller set of relevant words

Automated Speech Recognition

- Very small vocabulary for a given dialogue state → finite state grammar
- Larger vocabulary needed for dialogue state → n-gram language model with probabilities conditioned on the dialogue state
- State-specific language models are **restrictive grammars**
    - Few options for user → user has **less initiative**
    - More options for user → user has **more initiative**

Automated Speech Recognition

- ASR systems need to work quickly (users are often unwilling to wait for long pauses while their input is processed)
  - Prioritizing efficiency may necessitate constraining the vocabulary
- Generally return a confidence score for an output text sequence
  - Dialogue system can use this score to determine whether to request clarification, or move forward on the assumption that the sequence is correct



ˌmɪnɪˈsoʊtə

Hypothesis	Confidence
Minnesota	70%
Mini soda	30%

Minnesota

Natural Language Understanding

- Similar to the simple GUS frame-based architecture
  - Slot fillers are extracted from the user's utterance
- However, generally uses machine learning rather than rules

Dialogue State Tracker and Dialogue Policy

- **Dialogue State Tracker:** Maintains the current state of the dialogue
  - Most recent dialogue act
  - All slot values the user has expressed so far
- **Dialogue Policy:** Decides what the system should do or say next
  - In GUS, the system just asked questions until the frame was full
  - In more sophisticated dialogue systems, the policy might help the system decide:
    - When to answer the user's questions
    - When to ask the user a clarification question
    - When to make a suggestion

Natural Language Generation

- In GUS, sentences produced were from pre-written templates
- In more sophisticated dialogue systems, the natural language generation component can be **conditioned on prior context** to produce more natural-sounding dialogue turns

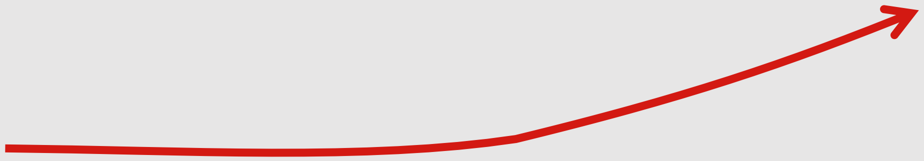
Text to Speech Synthesis

- Inputs:
    - Words
    - Prosodic annotations
- Output:
    - Audio waveform

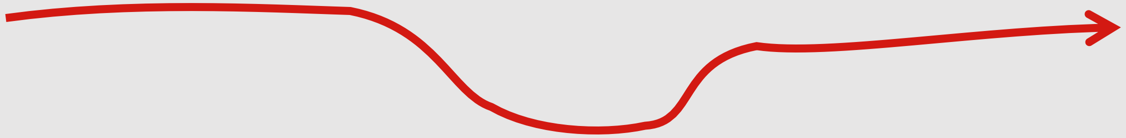
What is prosody?

- **Prosody:** Elements of speech such as **intonation**, **tone**, **stress**, and **rhythm**
- Often carries hints regarding:
  - A speaker's emotional state
  - The type of utterance being spoken
  - The presence of sarcasm
  - The focus of the utterance

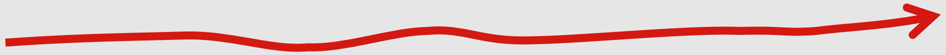
Common Prosodic Trends



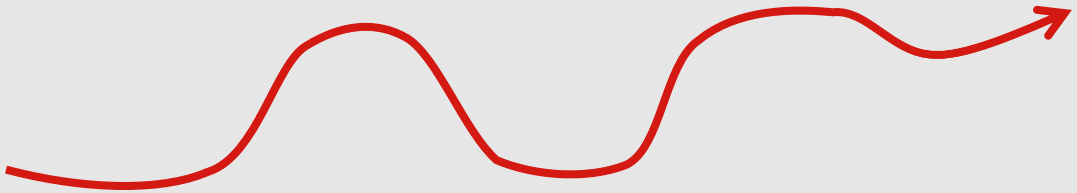
Question: Is there any popcorn left?



Sarcasm: I just love having no popcorn.



Anger: You ate my last bucket of popcorn!

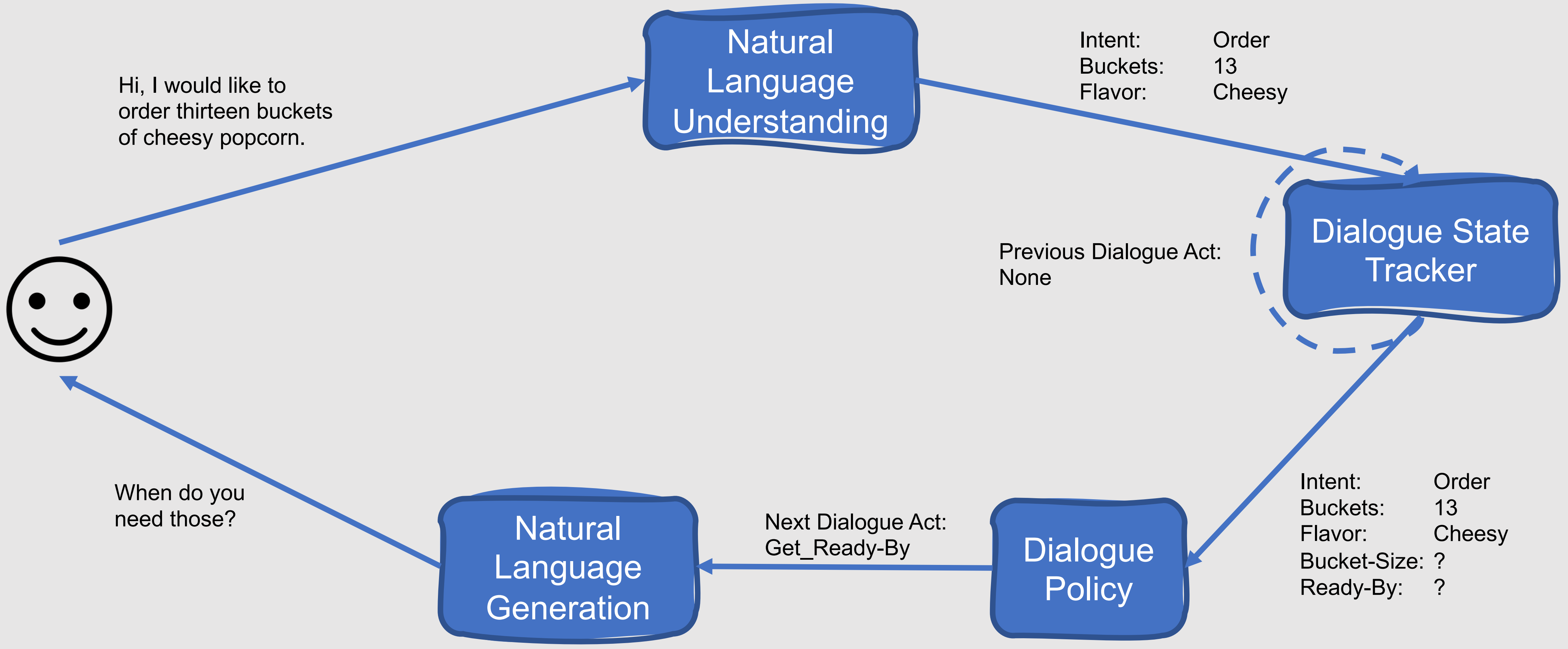


Emphasis: So, caramel or cheesy popcorn?

Spoken Dialogue Systems vs. Text-based Dialogue Systems

- Automated speech recognition and text to speech synthesis are only necessary in **spoken dialogue systems**
  - Dialogue systems which accept spoken input and produce spoken output
- Other dialogue systems can eliminate those components, moving directly from:
  - Input to natural language understanding
  - Natural language generation to output

Modified Dialogue State Architecture



Hi, I would like to order thirteen buckets of cheesy popcorn.

Natural Language Understanding

Intent: Order
Buckets: 13
Flavor: Cheesy

Dialogue State Tracker

Previous Dialogue Act:
None

Intent: Order
Buckets: 13
Flavor: Cheesy
Bucket-Size: ?
Ready-By: ?

Dialogue Policy

Next Dialogue Act:
Get_Ready-By

Natural Language Generation

When do you need those?

Dialogue Acts

- Dialogue-state systems make use of **dialogue acts**
  - Interactive function of the turn
    - Question
    - Statement
    - Clarification
- Different types of dialogue systems require that different types of dialogue acts are labeled
  - Dialogue tagsets tend to be task-specific

Sample Dialogue Act Tagset

Tag	Valid System Act?	Valid User Act?	Description
Hello(a=x, b=y, …)	🙂	🙂	Open a dialogue and give info a=x, b=y, …
Inform(a=x, b=y, …)	🙂	🙂	Give info a=x, b=y, …
Request(a, b=x, …)	🙂	🙂	Request value for a given b=x, …
Reqalts(a=x, …)		🙂	Request alternative with a=x, …
Confirm(a=x, b=y, …)	🙂	🙂	Explicitly confirm a=x, b=y, …
Confreq(a=x, …, d)	🙂		Implicitly confirm a=x, and request value of d
Select(a=x, a=y)	🙂		Implicitly confirm a=x, and request value of d
Affirm(a=x, b=y, …)	🙂	🙂	Affirm and give further info a=x, b=y, …
Negate(a=x)		🙂	Negate and give corrected value a=x
Deny(a=x)		🙂	Deny that a=x
Bye()	🙂	🙂	Close a dialogue

Sample Annotated Dialogue

Tag	Valid System Act?	Valid User Act?	Description
Hello(a=x, b=y, …)	🙂	🙂	Open a dialogue and give info a=x, b=y, …
Inform(a=x, b=y, …)	🙂	🙂	Give info a=x, b=y, …
Request(a, b=x, …)	🙂	🙂	Request value for a given b=x, …
Reqalts(a=x, …)		🙂	Request alternative with a=x, …
Confirm(a=x, b=y, …)	🙂	🙂	Explicitly confirm a=x, b=y, …
Confreq(a=x, …, d)	🙂		Implicitly confirm a=x, and request value of d
Select(a=x, a=y)	🙂		Implicitly confirm a=x, and request value of d
Affirm(a=x, b=y, …)	🙂	🙂	Affirm and give further info a=x, b=y, …
Negate(a=x)		🙂	Negate and give corrected value a=x
Deny(a=x)		🙂	Deny that a=x
Bye()	🙂	🙂	Close a dialogue

Speaker	Utterance	Dialogue Act
U	Hi, I am looking for somewhere to eat.	
S	You are looking for a restaurant. What type of food do you like?	
U	I'd like an Italian restaurant somewhere near the lake.	
S	Petterino's is a nice Italian restaurant near the lake.	
U	Is it reasonably priced?	
S	Yes, Petterino's is in the moderate price range.	
U	What is the phone number?	
S	The phone number for Peterino's is 123-456-7890	
U	Okay, thank you. Goodbye!	

Sample Annotated Dialogue

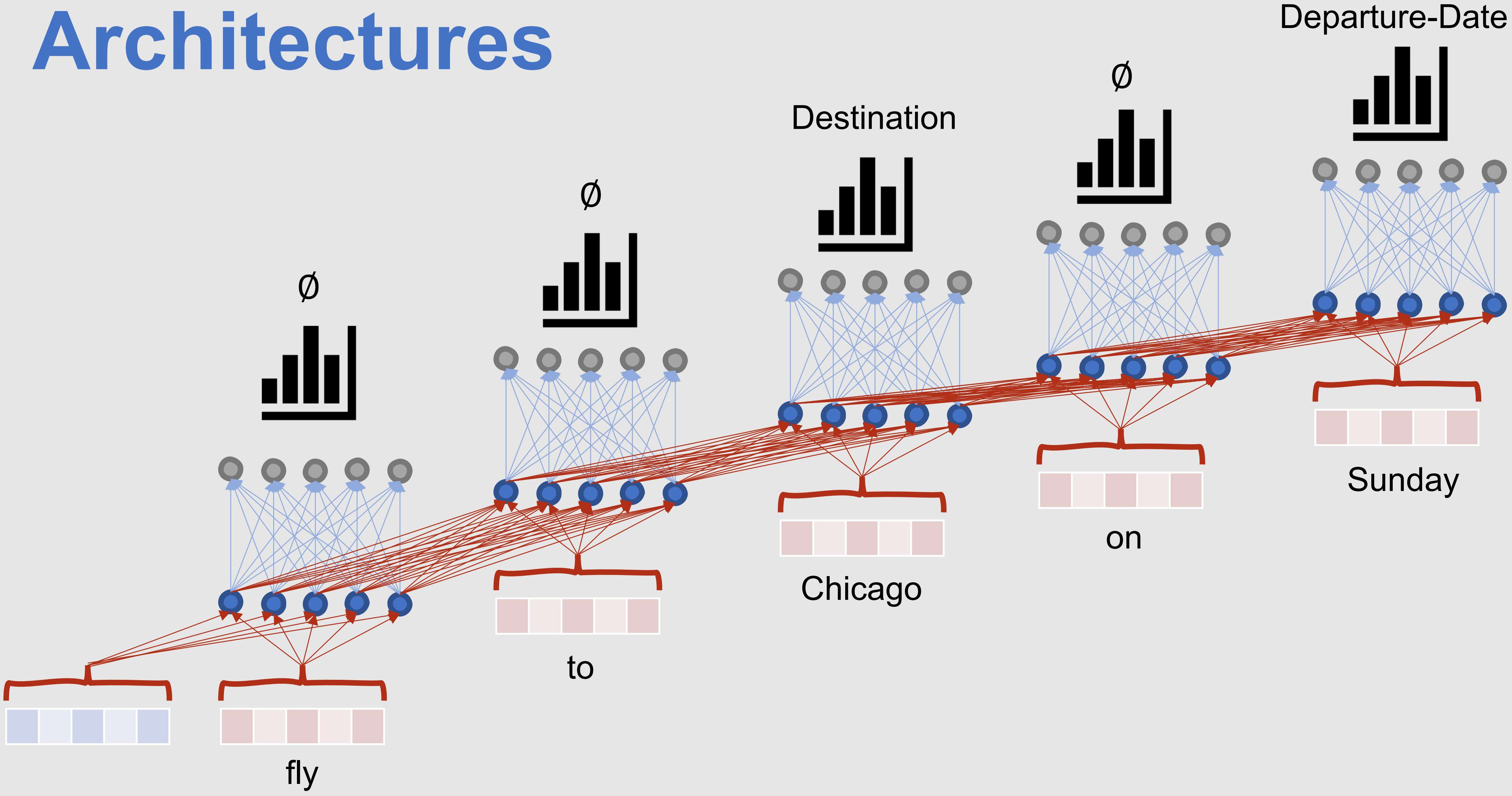
Tag	Valid System Act?	Valid User Act?	Description
Hello(a=x, b=y, …)	🙂	🙂	Open a dialogue and give info a=x, b=y, …
Inform(a=x, b=y, …)	🙂	🙂	Give info a=x, b=y, …
Request(a, b=x, …)	🙂	🙂	Request value for a given b=x, …
Reqalts(a=x, …)		🙂	Request alternative with a=x, …
Confirm(a=x, b=y, …)	🙂	🙂	Explicitly confirm a=x, b=y, …
Confreq(a=x, …, d)	🙂		Implicitly confirm a=x, and request value of d
Select(a=x, a=y)	🙂		Implicitly confirm a=x, and request value of d
Affirm(a=x, b=y, …)	🙂	🙂	Affirm and give further info a=x, b=y, …
Negate(a=x)		🙂	Negate and give corrected value a=x
Deny(a=x)		🙂	Deny that a=x
Bye()	🙂	🙂	Close a dialogue

Speaker	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 restaurant somewhere near the lake.	Inform(food=Italian, near=lake)
S	Petterino's is a nice Italian restaurant near the lake.	Inform(name="Petterino's", type=restaurant, food=Italian, near=lake)
U	Is it reasonably priced?	Confirm(pricerange=moderate)
S	Yes, Petterino's is in the moderate price range.	Affirm(name="Petterino's", pricerange=moderate)
U	What is the phone number?	Request(phone)
S	The phone number for Peterino's is 123-456-7890	Inform(name="Petterino's", phone="123-456-7890")
U	Okay, thank you. Goodbye!	Bye()

Slot Filling in Dialogue State Architectures

- Special case of supervised semantic parsing
  - Labeled training set associates each sentence with the correct set of slots, domain, and intent
- Many possible ways train a classifier for this purpose
- One method:
  - Train a sequence model to map from input words to slot fillers, domain, and intent

Slot Filling in Dialogue State Architectures



Slot Filling in Dialogue State Architectures

- Domain and intent can be determined via:
  - One vs. many classifier
  - Adding domain+intent as the desired output for the final end-of-sentence token in the sequence labeler

Common Industrial Approach to ML-based Slot Filling

- **Bootstrapping!**
- Start with a seed GUS-style, rule-based system
- Predict labels for new user utterances
- Train classifier on new utterances with predicted labels
- Repeat as needed

Dialogue Management

- Core component of task-based dialogue systems
  - Decides what step to take next to bring the conversation closer to its goal
- Can range from simple (minimal history and/or state tracking) to complex (advanced state tracking and dialogue policy modules)
- Simplest dialogue management architecture:
  - **Finite state dialogue manager**

Finite State Dialogue Manager

States (nodes)

- Questions that the dialogue manager asks the user

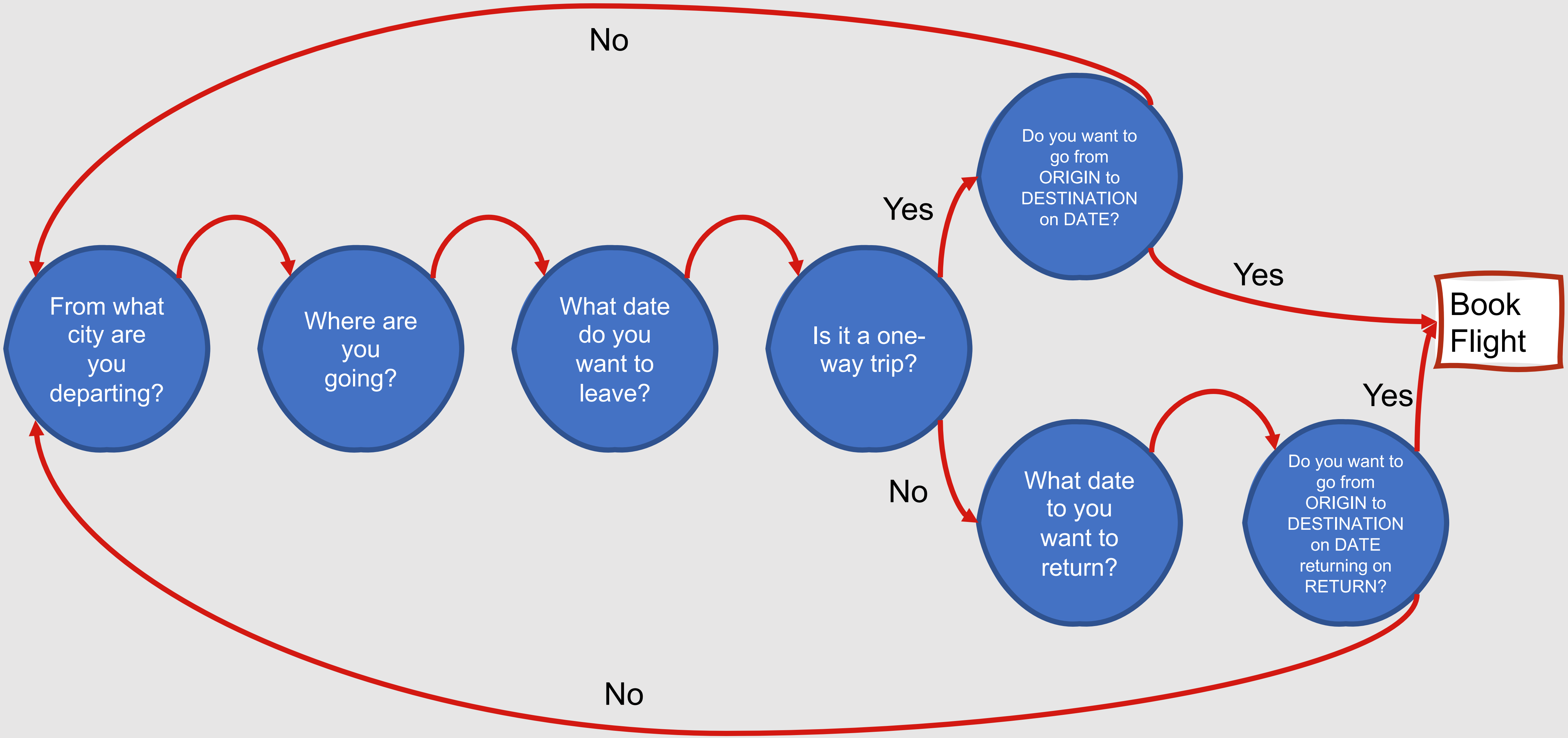
Transitions (arcs)

- Actions to take depending on how the user responds

System has full conversational initiative!

- Asks a series of questions
- Ignores or misinterprets inputs that are not direct answers to questions

Finite State Dialogue Manager


No
Do you want to go from ORIGIN to DESTINATION on DATE?
Yes
Yes
From what city are you departing?
Where are you going?
What date do you want to leave?
Is it a one-way trip?
Book Flight
Yes
No
What date to you want to return?
Do you want to go from ORIGIN to DESTINATION on DATE returning on RETURN?
No

Finite State Dialogue Manager

- Many finite state systems also allow **universal commands**
  - Commands that can be stated anywhere in the dialogue and still be recognized
    - Help
    - Start over
    - Correction
    - Quit

Advantages and Disadvantages of Finite State Dialogue Managers

Advantages:	Disadvantages:
• Easy to implement • Sufficient for simple tasks	• Can be awkward and annoying • Cannot easily handle complex sentences

Dialogue Management

- More common in modern dialogue state architectures:
  - **Dialogue state tracker**
  - **Dialogue policy**

What does a dialogue state tracker do?

Determine both:

- The current state of the frame
  - What slots have been filled, and how?
- The user's most recent dialogue act

Current state of the frame: More than just the slot fillers expressed in the current sentence!

- Entire state of the frame up to and including this point

Example: Dialogue State Tracker

I'm looking for an upscale restaurant.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

Example: Dialogue State Tracker

I’m looking for an upscale restaurant.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

inform(price=expensive)

Example: Dialogue State Tracker

I’m looking for an upscale restaurant.

Sure. What cuisine?

Turkish food would be great.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

inform(price=expensive)

Example: Dialogue State Tracker

I'm looking for an upscale restaurant.

Sure. What cuisine?

Turkish food would be great.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

inform(price=expensive,
cuisine=Turkish)

Example: Dialogue State Tracker

I'm looking for an upscale restaurant.

Sure. What cuisine?

Turkish food would be great.

Okay. Where should the restaurant be?

Near the Chicago Theatre.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

inform(price=expensive,
cuisine=Turkish,
area=ChicagoTheatre)

Example: Dialogue State Tracker

I’m looking for an upscale restaurant.

Sure. What cuisine?

Turkish food would be great.

Okay. Where should the restaurant be?

Near the Chicago Theatre.

So an upscale Turkish restaurant near the Chicago Theatre?

Yes, please.

Tag
Hello(a=x, b=y, …)
Inform(a=x, b=y, …)
Request(a, b=x, …)
Reqalts(a=x, …)
Confirm(a=x, b=y, …)
Confreq(a=x, …, d)
Select(a=x, a=y)
Affirm(a=x, b=y, …)
Negate(a=x)
Deny(a=x)
Bye()

Dialogue State Tracker

inform(price=expensive,
cuisine=Turkish,
area=ChicagoTheatre);

affirm(price=expensive,
cuisine=Turkish,
area=ChicagoTheatre)

How can we detect corrections?

- Users generally correct errors (either theirs or the system's) by **repeating** or **reformulating** their utterance
- Harder to do than detecting regular utterances!
  - Speakers often **hyperarticulate** corrections
- Common characteristics of corrections:
  - Exact or close-to-exact repetitions
  - Paraphrases
  - Contain "no" or swear words
  - Low ASR confidence

Turkish food sounds great---or, wait, no. How about Greek food instead?

I said TURKISH FOOD not TURKEY LEGS.

No. TURK-ISH-FOOD!

Dialogue Policy

- Goal: Determine **what action the system should take next**
  - What dialogue act should be generated?
- More formally, at turn *i* in the conversation, we want to predict which action A_i to take, **based on the entire dialogue state** (the sequence of dialogue acts from the system *A* and the user *U*):
  - $A_i' = \underset{A_i \in A}{\operatorname{argmax}} P(A_i | A_1, U_1, \ldots, A_{i-1}, U_{i-1})$
- To simplify this, we can maintain the dialogue state as the set of slot-fillers the users has expressed (thereby allowing us to condition on **the current state of *Frame*$_i$ and the last turn**):
  - $A_i' = \underset{A_i \in A}{\operatorname{argmax}} P(A_i | Frame_{i-1}, A_{i-1}, U_{i-1})$

How can we estimate these probabilities?

- **Neural classifier** trained on **vector representations of the slot fillers and utterances**
- More sophisticated models may also use **reinforcement learning**
  - Reinforcement learning system gets a reward at the end of a dialogue
  - Uses that reward to train a policy to take an optimal sequence of actions
    - Large positive reward if the dialogue system terminates with the correct slot representation for a training instance
    - Large negative reward if all slots are wrong

How can dialogue managers handle mistakes?

- First, check to make sure the user’s input has been interpreted correctly:
  - **Confirm understandings** with the user
  - **Reject utterances** that the system is likely to have **misunderstood**
- These checks can be performed **explicitly** or **implicitly**

Explicit Confirmation

- System asks the user a direct question to confirm its understanding

S: From which city do you want to leave?

U: Chicago.

S: You want to leave from Chicago?

U: Yes.

U: I'd like to fly from Chicago to Dallas on November twenty-seventh.

S: Okay, I have you going from Chicago to Dallas on November twenty-seventh. Is that correct?

U: Yes.

Implicit Confirmation

- System demonstrates its understanding as a **grounding** strategy
- Usually done by repeating back its understanding as part of the next question

U: I want to travel to Chicago.

S: When do you want to travel to Chicago?

U: Hi, I'd like to fly to Chicago tomorrow afternoon.

S: Traveling to Chicago on November fifteenth in the afternoon. What is your full name?

When to use explicit vs. implicit confirmation?

Explicit Confirmation

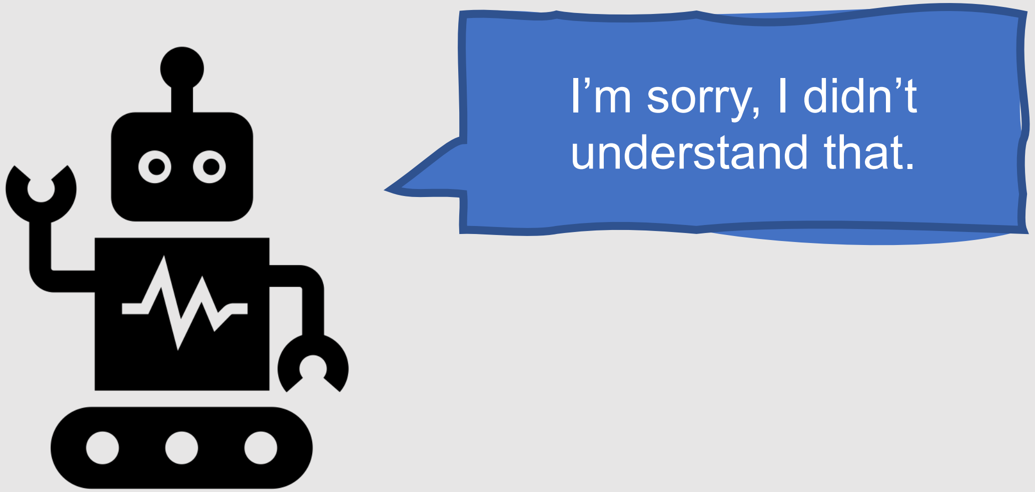
- Easier for users to correct the system’s misrecognitions
    - User can just say “no” when the system tries to confirm
- But, can be awkward and unnatural
- Also extends the length of the conversation

Implicit Confirmation

- Sounds more conversationally natural
- More efficient
- But, more difficult for users to correct the system’s misrecognitions

What if the dialogue manager has no idea how to handle the user's input?

- Systems can outright **reject** the user's input
- Often, when utterances are rejected, systems will follow a strategy of **progressive prompting** or **escalating detail**



Progressive Prompting

- System gradually gives the user more guidance about how to formulate an utterance that will be accepted

S: When would you like to leave?

U: Well, I need to be in Dallas for Thanksgiving.

S: Sorry, I didn't get that. Please say the month and day you'd like to leave.

U: I would like to leave on November twenty-seventh.

Other Strategies for Error Handling

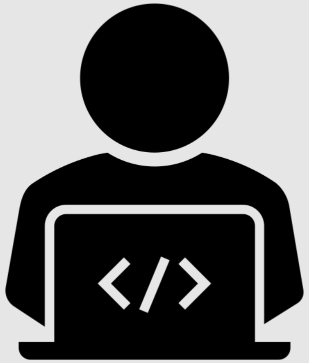
- **Rapid Reprompting:** For the first rejection, the system just says “I’m sorry?” or “What was that?”
  - From the second rejection onward, **progressive prompting** can be applied
  - **Users tend to prefer rapid reprompting** as a first-level error prompt (Cohen et al., 2004)
- Explicitly confirm **low-confidence ASR outputs**
- **Design thresholds based on the cost** of making an error
  - Low confidence → Reject
  - Confidence just above minimum threshold → Confirm explicitly
  - Confidence comfortably above threshold → Confirm implicitly
  - Very high confidence → Don’t confirm at all

Natural Language Generation

- Creates a natural language realization of the selected dialogue act
- Two stages:
  - **Content planning**
    - What should be said?
  - **Surface realization**
    - How should it be said?

Content Planning

- Most of the work is done by the dialogue policy
  - **Which dialogue act** should be generated?
  - **What attributes** (slots and values) should be included in the dialogue act?



Dialogue Act: Recommend

Prespecified Attributes:
Cuisine=Turkish
Area=ChicagoTheatre
Price=Expensive

Surface Realization

- Generates a sentence of the **specified type**, containing the **specified attributes**
- Often a machine learning model trained on many examples of representation/sentence pairs
  - Recommend(Cuisine=Turkish, Area=ChicagoTheatre, Price=Expensive)
    - “So you want an upscale Turkish restaurant near the Chicago Theatre?”
    - “Okay, so we’re looking for a high-end Turkish restaurant near the Chicago Theatre.”

What if it is hard to find training data like this?

- Unlikely that we'll see every possible combination of attributes
- Thus, generality of training samples can be increased by **delexicalization**
- **Delexicalization:** The process of replacing **specific words in the training set that represent slot values** with **generic placeholder tokens**

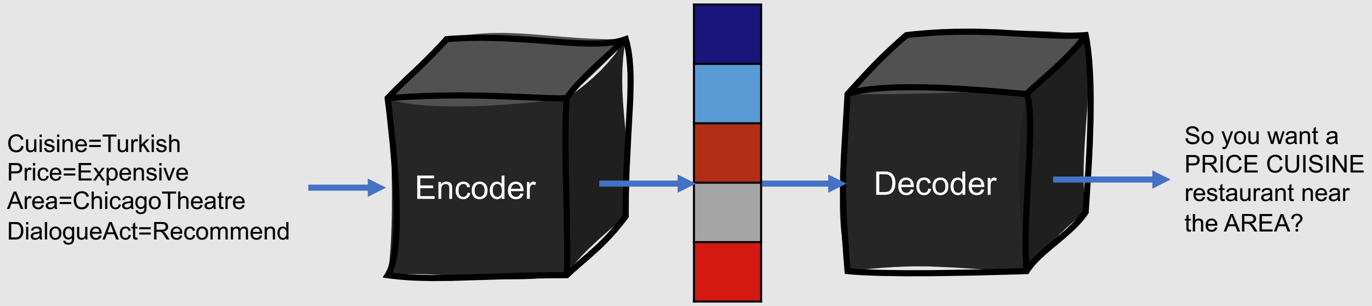
Recommend(Cuisine=Turkish, Area=ChicagoTheatre, Price=Expensive)

So you want an PRICE CUISINE restaurant near the AREA?

Okay, so we're looking for a PRICE CUISINE restaurant near the AREA.

Mapping from Frames to Delexicalized Sentences

- Generally performed by encoder-decoder models
- Input: Sequence of tokens that represent the dialogue act and its arguments
    - Cuisine=Turkish
    - Price=Expensive
    - Area=ChicagoTheatre
    - DialogueAct=Recommend
- Output: Delexicalized sentence

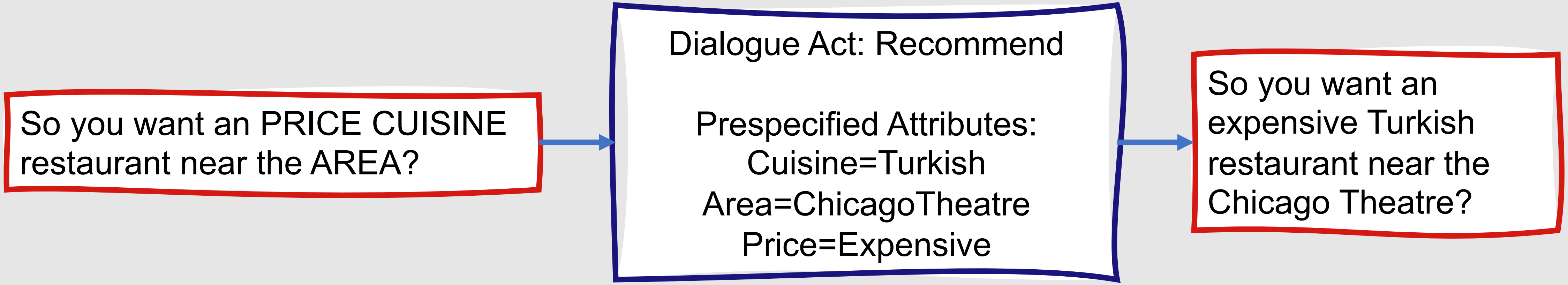


Cuisine=Turkish
Price=Expensive
Area=ChicagoTheatre
DialogueAct=Recommend

So you want a PRICE CUISINE restaurant near the AREA?

Relexicalization

- Once we've generated a delexicalized string, we need to **relexicalize** it
- **Relexicalization:** Filling in **generic slots** with **specific words**
- We can do this using the input frame from the content planner



So you want an PRICE CUISINE restaurant near the AREA?

Dialogue Act: Recommend

Prespecified Attributes:
Cuisine=Turkish
Area=ChicagoTheatre
Price=Expensive

So you want an expensive Turkish restaurant near the Chicago Theatre?

Generating Clarification Questions

- Humans clarification questions tend to be targeted at specific elements of the misunderstanding (Purver 2004, Ginzburg and Sag 2000, Stoyanchev et al. 2013)
- These can be created using rule-based approaches, or by probabilistically guessing which slots in a sentence might have been misrecognized

What flights are going to UNKNOWN on November 14th?

What flights are going to where on November 14th?

Can you please repeat your destination for November 14th?

Evaluating Task-based Dialogue Systems

- Task success
- User satisfaction
- Efficiency cost
- Quality cost

Measuring Task Success

- How correct was the total solution?
- **Slot Error Rate:** The percentage of slots that were filled with incorrect values
  - $\text{Slot Error Rate} = \dfrac{\text{\# of inserted, deleted, or substituted slots}}{\text{\# of total reference slots}}$

What flights are going from Chicago to Dallas on the afternoon of November 27th?

Slot	Filler
ORIGIN	Chicago
DESTINATION	Denver
TIME	afternoon
DATE	11/27/2019

Slot Error Rate = ¼ = 0.25

Measuring Task Success

- Alternative metric: **task error rate**
- **Task Error Rate:** The percentage of times that the overall task was completed incorrectly
  - Was the (correct) meeting added to the calendar?
  - Did users end up booking the flights they wanted?
- In addition to **slot error rate** and **task error rate**, we can apply our standard NLP metrics:
  - **Precision**
  - **Recall**
  - **F-measure**

Measuring User Satisfaction

- Typically survey-based
- Users interact with a dialogue system to perform a task, and then complete a questionnaire about their experience

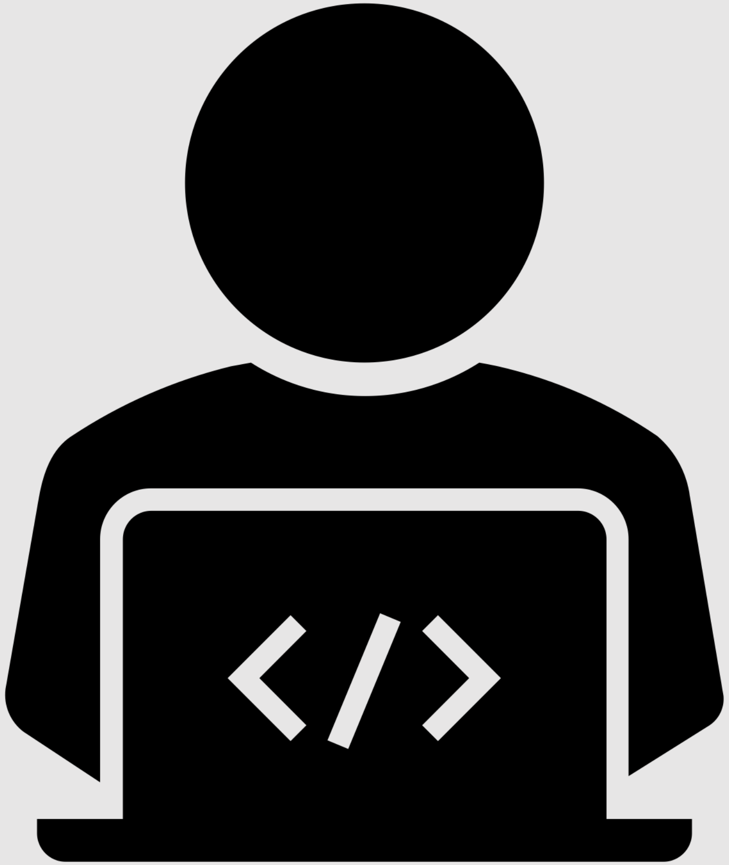
On a scale from 1 (worst) to 5 (best)….	
TTS Performance	Was the system easy to understand?
ASR Performance	Did the system understand what you said?
Task Ease	Was it easy to find the information you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
User Expertise	Did you know what you could say at each point?
System Response	Was the system often sluggish and slow to reply to you?
Expected Behavior	Did the system work the way you expected it to?
Future Use	Do you think you'd use the system in the future?

Measuring Efficiency Cost

- How efficiently does the system help users perform tasks?
- Total **elapsed time**
- Number of **total turns**
- Number of **system turns**
- Number of **user queries**
- **Turn correction ratio**
  - Number of system or user turns that were used solely to correct errors, divided by the total number of turns

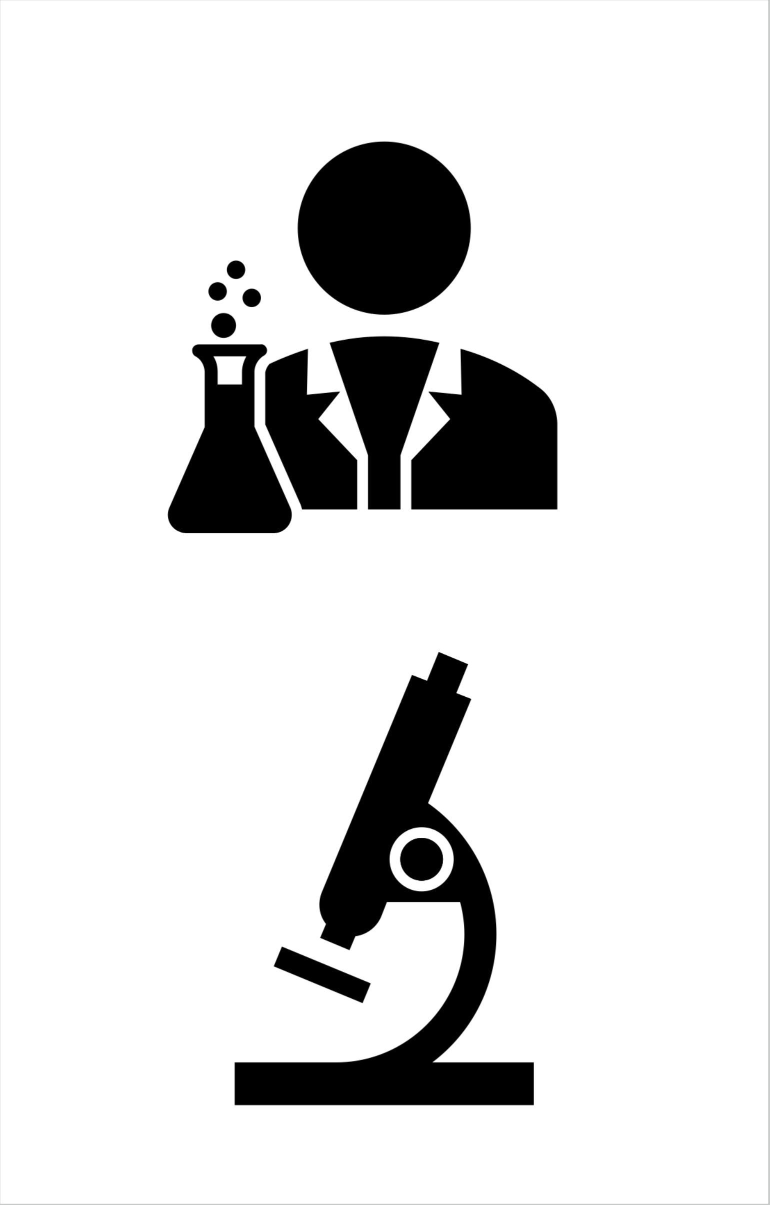
Measuring Quality Cost

- What are the costs of other aspects of the interaction that affect users' perceptions of the system?
- Number of **times the ASR system fails** to return anything useful
- Number of **times the user had to interrupt** the system
- Number of **times the user didn't respond** to the system quickly enough (causing event time-outs or follow-up prompts)
- **Appropriateness/correctness** of the system's questions, answers, and error messages



Dialogue System Design

- Users play an important role in designing dialogue systems
  - Research in dialogue systems is closely linked to research in **human-computer interaction**
- Design of dialogue strategies, prompts, and error messages is often referred to as **voice user interface design**



Voice User Interface Design

- Generally follows **user-centered design principles**
  1. Study the user and task
  2. Build simulations and prototypes
  3. Iteratively test the design on users

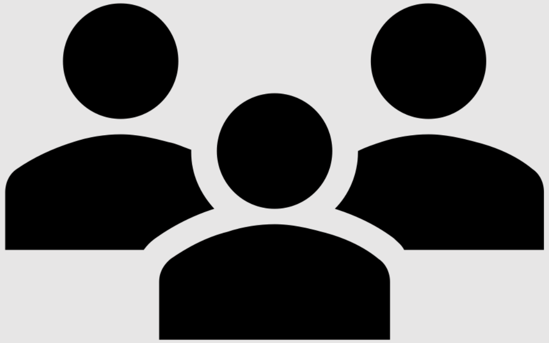
Studying the User and Task

- **Understand the potential users**
  - Interview them about their needs and expectations
  - Observe human-human dialogues
- **Understand the nature of the task**
  - Investigate similar dialogue systems
  - Talk to domain experts

Building Simulations and Prototypes

- **Wizard-of-Oz Studies:** Users interact with what they *think* is an automated system (but that is actually a human "wizard" disguised by a software interface)
- Wizard-of-Oz studies can be used to **test architectures** prior to implementation
  1. Wizard gets input from the user
  2. Wizard uses a database to run sample queries based on the user input
  3. Wizard outputs a response, either by typing it or by selecting an option from a menu
  4. Often used in text-only interactions, but the output can be disguised using a text to speech system for voice interfaces
- Wizard-of-Oz studies can also be used to **collect training data**
- Although not a perfect simulation of the real system (they tend to be idealistic), results from Wizard-of-Oz studies provide a useful first idea of **domain issues**

Iteratively Testing the Design



- Often, users will interact with the system in unexpected ways
- Testing prototypes early (and often) minimizes the chances of substantial issues in the final version
  - Application designers are often not able to anticipate these issues since they've been working on the design for so long themselves!

Ethical Issues in Dialogue System Design

- **Bias**
  - Machine learning systems of any kind tend to replicate human biases that occur in training data
  - Especially problematic for chatbots that are trained to replicate human responses!
- Microsoft's Tay chatbot: https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist
- Corpora drawn from social media (e.g., Twitter or Reddit) tend to be particularly problematic (Henderson et al. 2017, Hutto et al. 2015, Davidson et al. 2017)

Ethical Issues in Dialogue System Design

- **Privacy**
  - Home dialogue agents may accidentally record private information, which may then be used to train a conversational model
  - Adversaries can potentially recover this information
    - Very important to anonymize personally identifiable information when training chatbots on transcripts of human-human or human-machine conversation!
- **Gender Equality**
  - Current chatbots tend to be assigned female names and voices
    - Perpetuates stereotypes of subservient females
    - Most commercial chatbots evade or give positive responses to sexually harassing language, rather than responding in clear negative ways (Fessler, 2017)

Summary: Dialogue Systems and Chatbots (Part 2)

- Modern dialogue systems tend to use the **dialogue-state architecture**, which contains components for:
    - **Automated speech recognition**
    - **Natural language understanding**
    - **Dialogue state tracking**
    - **Dialogue policy**
    - **Natural language generation**
    - **Text to speech**
- These components have to handle many expected and unexpected inputs (**different dialogue act types**, as well as **unrecognized**, **corrected**, or **mistaken** input)
- Dialogue systems are typically evaluated based on **task success**, **user satisfaction**, **efficiency cost**, and **quality cost**
- One way to gain an initial understanding of domain issues (as well as to collect relevant data) is to conduct a **Wizard-of-Oz** study
- Dialogue system designers should be aware of ethical issues in dialogue system design, including concerns about **bias**, **privacy**, and **gender equality**