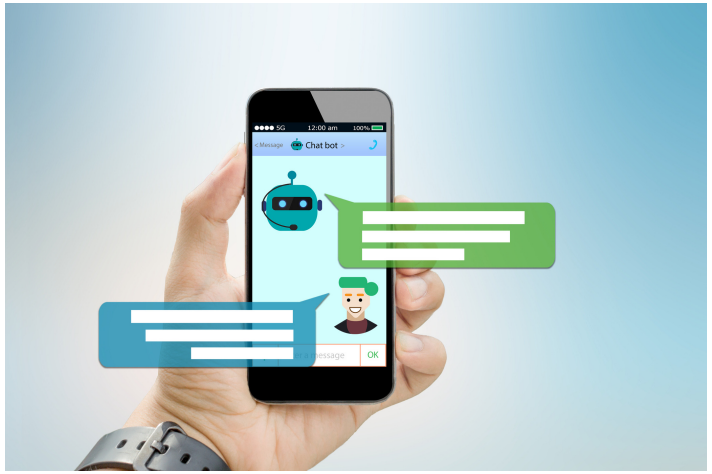
Dialogue Systems and Chatbots

Natalie Parde

UIC CS 421



What is a dialogue system?

- Broadly speaking, a program that can communicate with users
  - This may be through speech, text, or both
- Often also referred to as **chatbots** or **conversational agents**

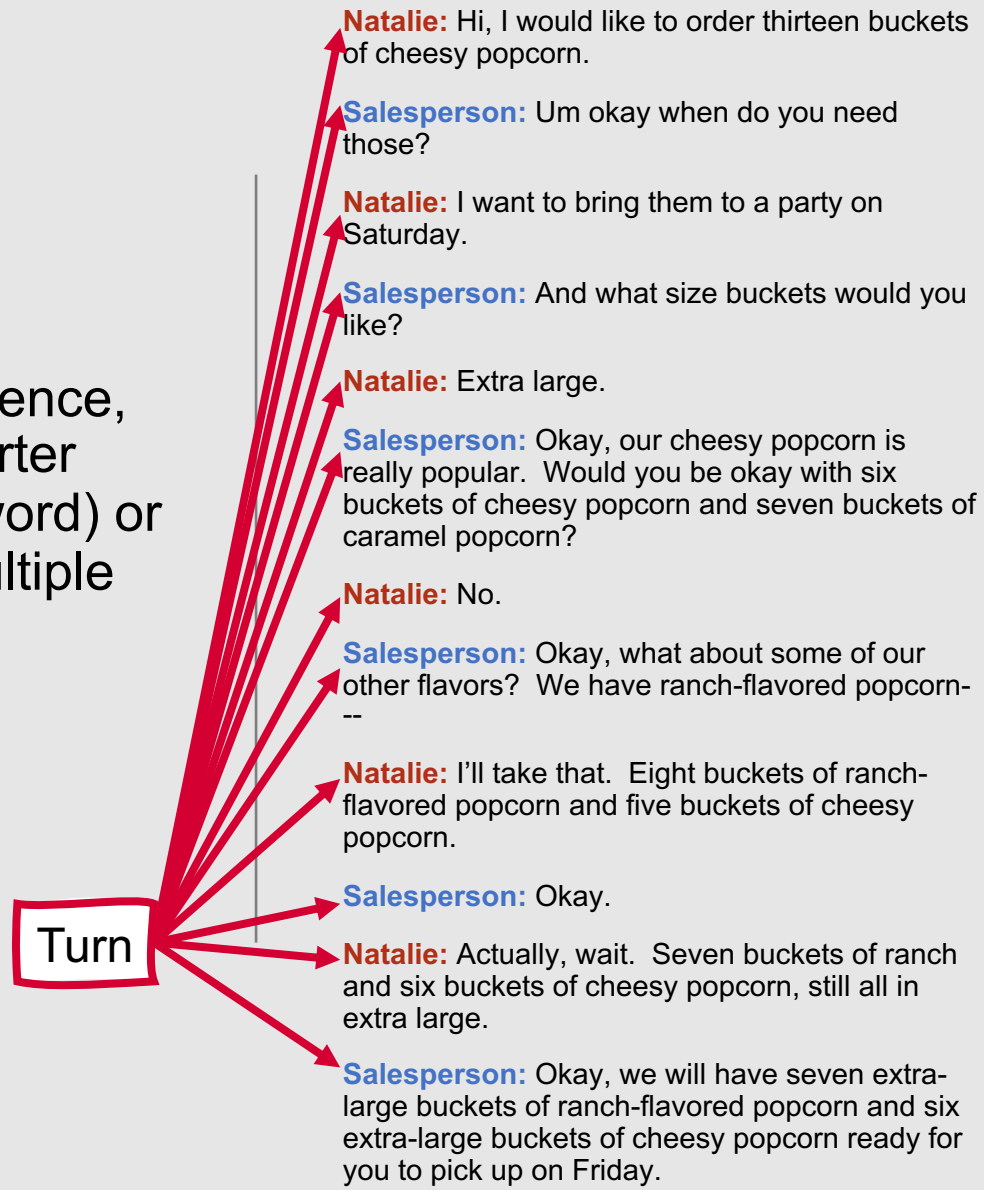
Types of Dialogue Systems

- **Task-Oriented:** Designed to leverage conversational interactions to help users complete tasks
- **Conversational Chatbots:** Designed to carry out extended, unstructured conversations (similar to human chats)
- Many dialogue systems contain elements of both categories

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

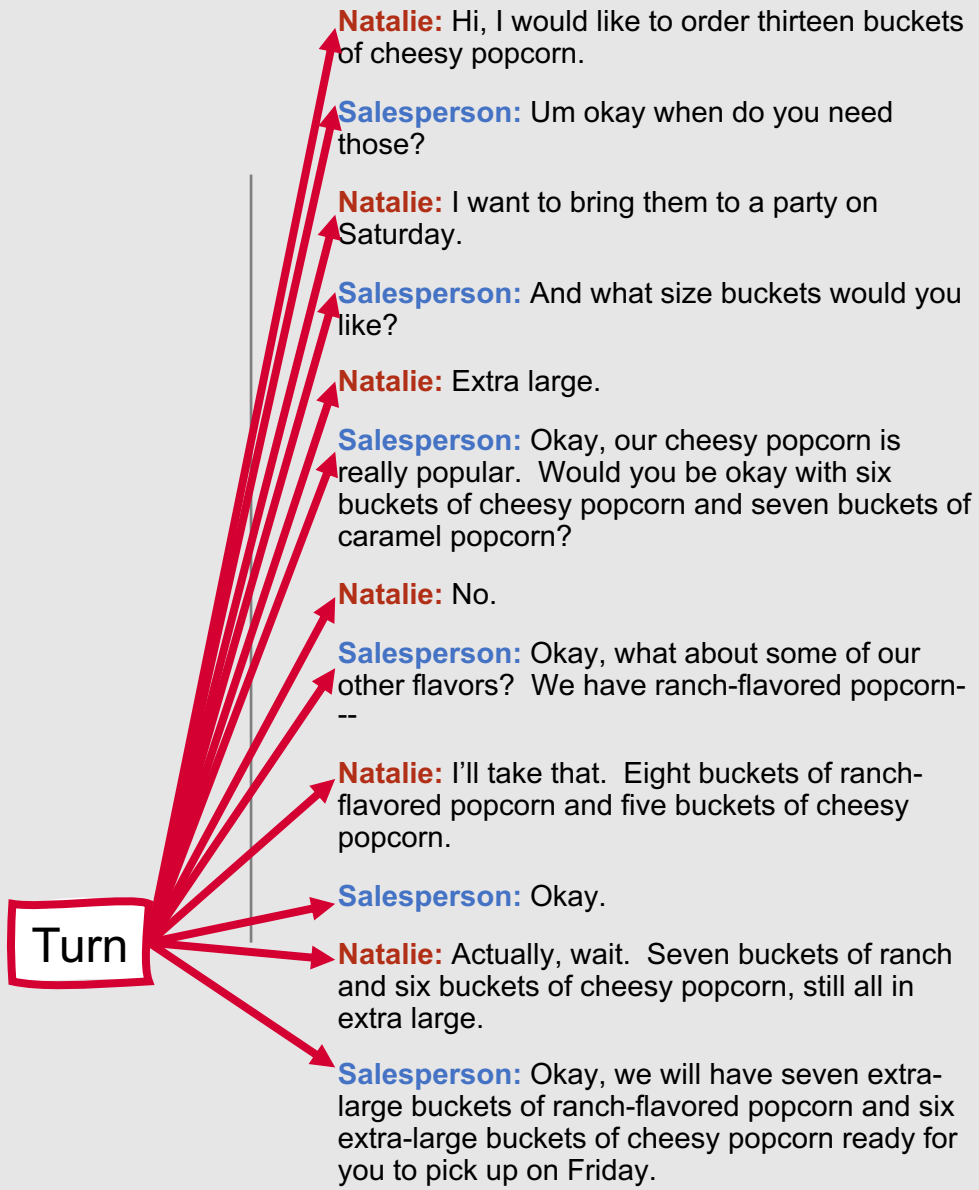
Properties of Human Conversation

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



Properties of Human Conversation

- Understanding turn structure is very important for spoken dialogue systems!
- Systems must perform accurate **endpoint detection**:
  - When to stop talking
  - When to start talking



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.

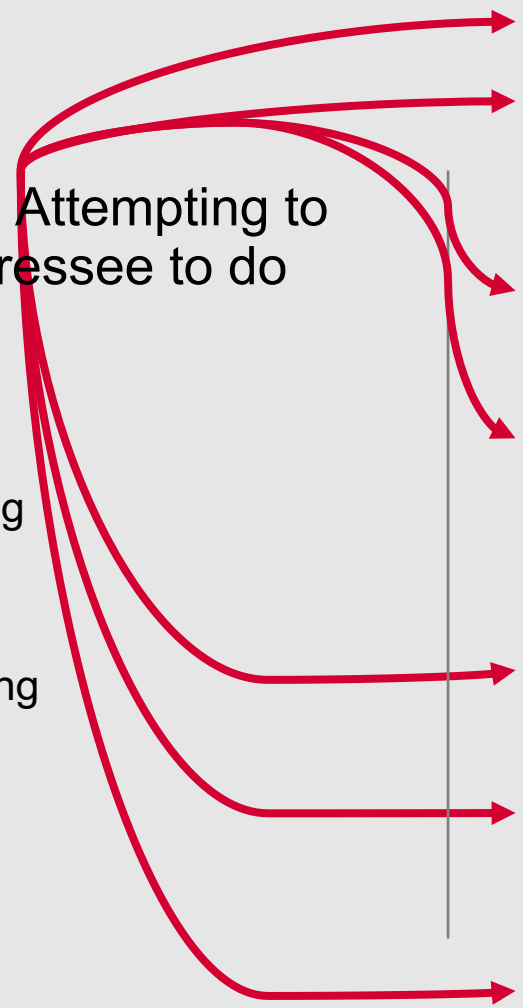
Salesperson: Okay.

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.

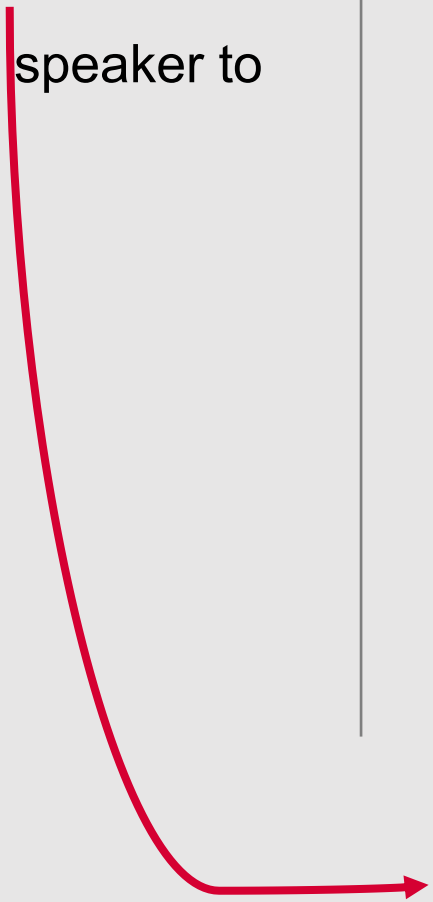
Salesperson: Okay.

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

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.

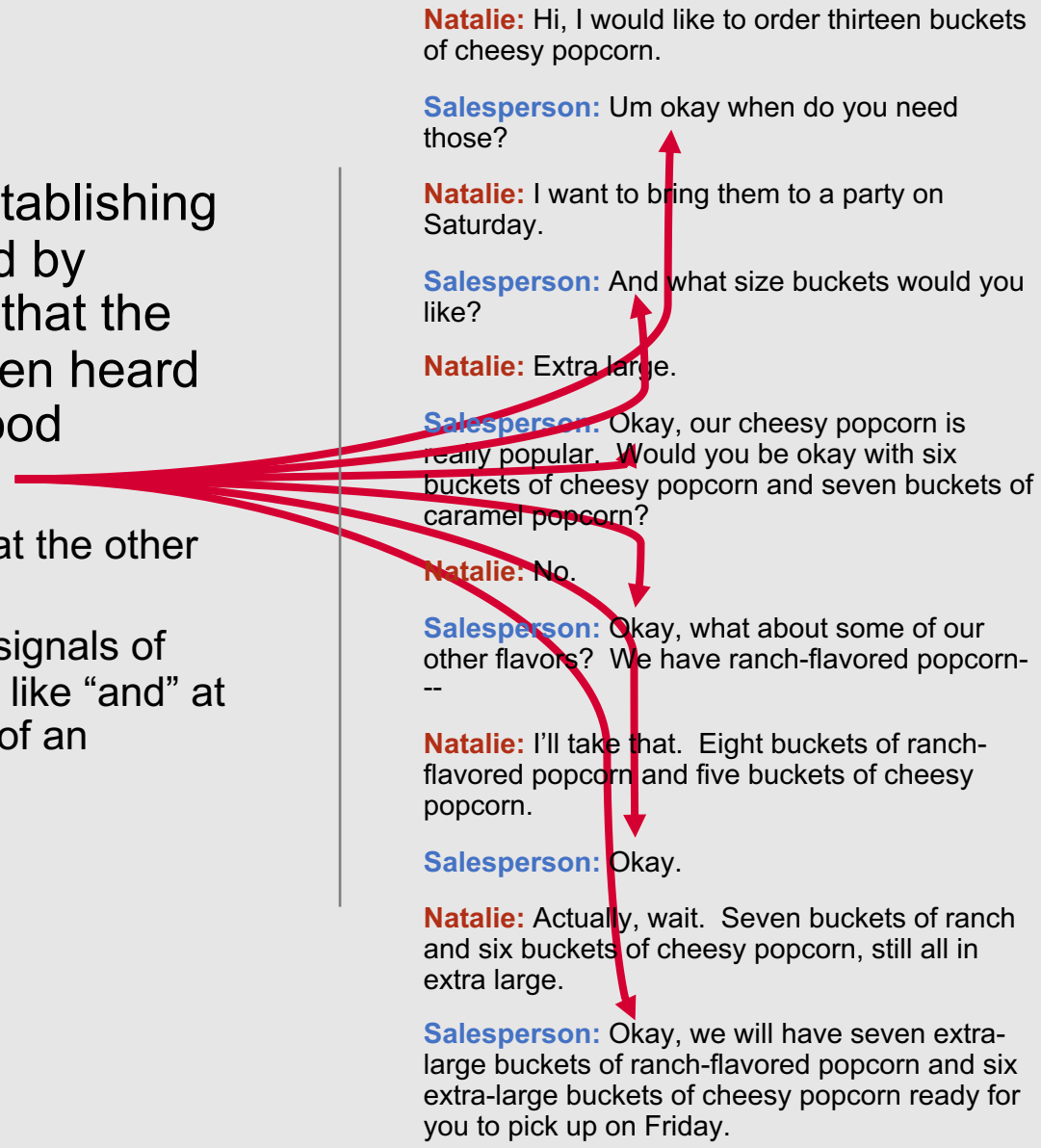
Salesperson: Okay.

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



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
- Adjacency pairs are dialogue acts that naturally appear together
  - First pair part: Question
  - Second pair part: Answer
- They can be separated by side sequences or subdialogues

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

- 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!
- Question answering systems (e.g., “Alexa, what’s the weather like right now?”) are **user-initiative** systems
- In contrast, **system-initiative** can be 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.

Back to chatbots….

- Chatbots carry out extended conversations with the **goal of mimicking unstructured, informal, human-human chats**
- They can be used for **entertainment** or **practical** purposes
- Common Architectures:
  - **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

Example: 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 to know nothing to fulfill a 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 ranked **keyword(s)** that might occur in users' sentences

Rule-based Chatbots

- What if no keywords are matched in an input?
  - ELIZA accesses facts from memory or defaults to a non-committal response
    - PLEASE GO ON
    - THAT'S VERY INTERESTING
    - I SEE

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)

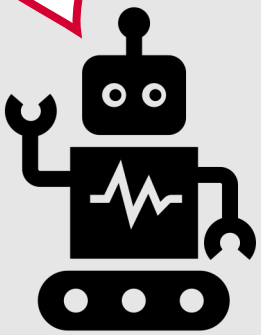
Men are all alike.

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)

Men are all alike.

IN WHAT WAY

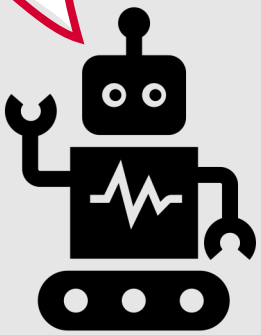


Example: ELIZA

They're always bugging us about something or other.

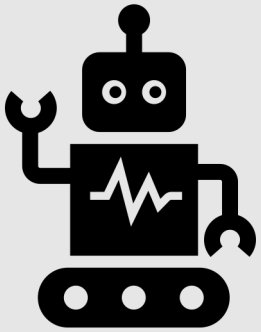
Rule	Rank
(I .*) → (YOU SAY YOU 2)	2
(I'm .*) → (I AM SORRY TO HEAR THAT YOU ARE 2)	1
(my .*) → (YOUR 2)	1
(.* always .*) → CAN YOU THINK OF A SPECIFIC EXAMPLE	1
(.*) → IN WHAT WAY	3
(.* my .*) → (DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3)	(Memory)

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)



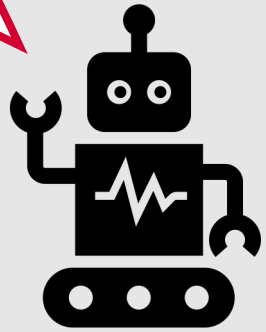
Example: ELIZA

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)

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

YOUR BOYFRIEND MADE YOU COME HERE



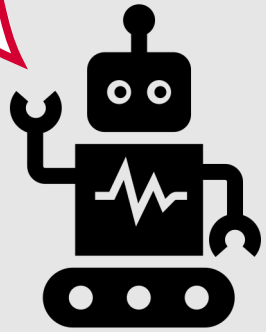
Example: ELIZA

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)

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

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

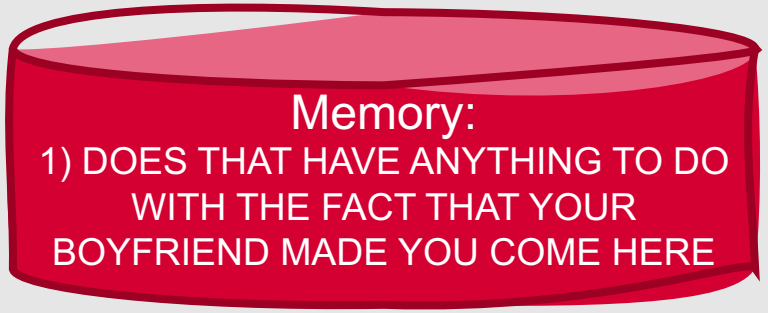


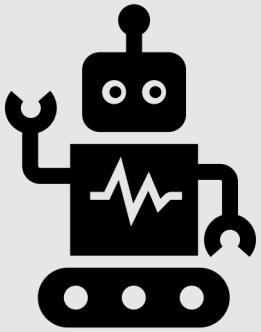
Example: ELIZA

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)



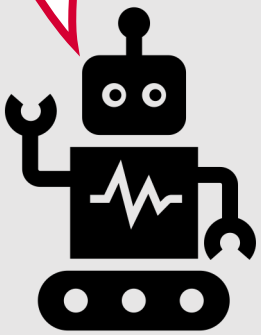


Example: ELIZA

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)

Maybe he is the problem.

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



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
  - 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

- No manually created rules
- Instead, learn mappings from inputs to outputs based on large human-human conversation corpora
- Very data-intensive!
  - May require hundreds of millions, or even billions, of words

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

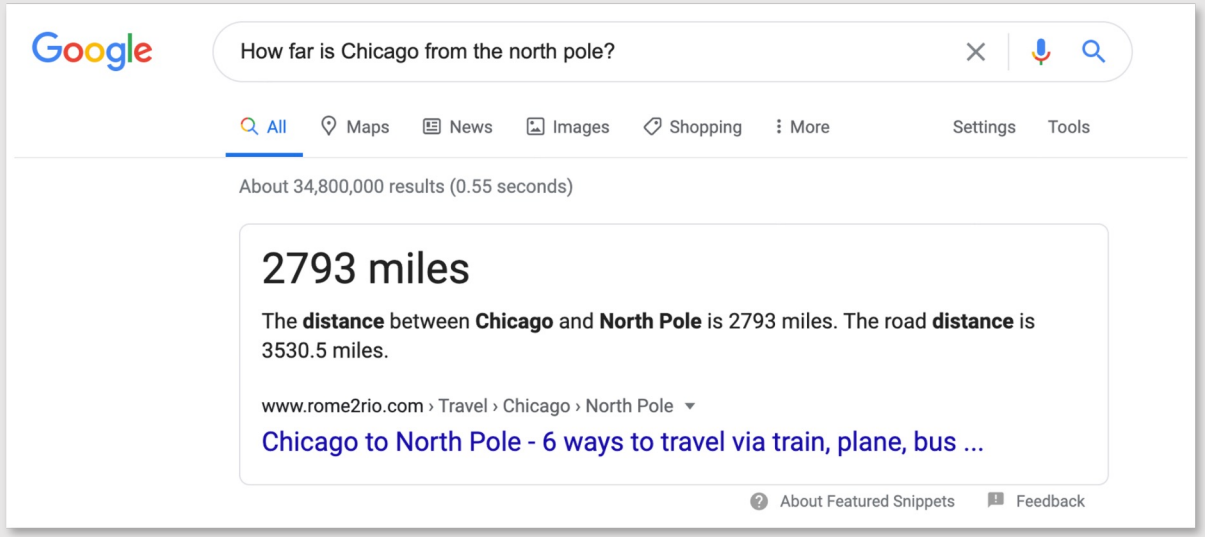
- Large spoken conversational corpora
  - Switchboard corpus of American English telephone conversations: https://catalog.ldc.upenn.edu/LDC97S62
- Movie dialogue
- Text from microblogging sites (e.g., Twitter)
- Collections of crowdsourced conversations
  - Topical-Chat: https://github.com/alexa/alexa-prize-topical-chat-dataset
- News or online knowledge repositories
- Collected user input
  - Beware of privacy concerns!

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)

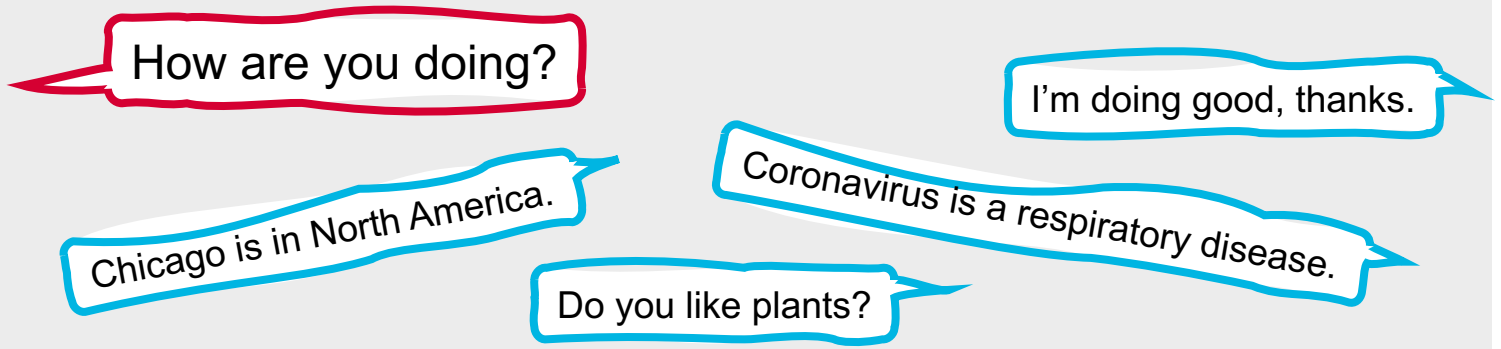
Corpus-based Chatbots

- This makes most corpus-based chatbots similar to **question answering systems**:
  - Focus on single responses
  - Ignore 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
- Two simple information retrieval methods for choosing appropriate responses:
  - Return the response to the most similar turn
  - Return the most similar turn



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

- Incorporate 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**

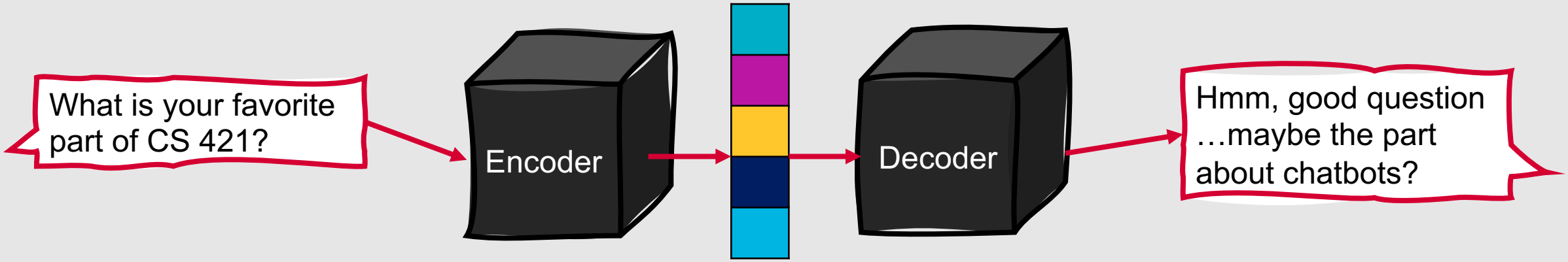
$$\underset{t \in C}{\operatorname{argmax}} \frac{q^T t}{\|q\| \|t\|}$$

Encoder-Decoder Chatbots

- **Machine learned sequence transduction:** System learns from a corpus to **transduce a question to an answer**
  - Machine learning version of ELIZA
  - **Encoder-decoder** models accept sequential information as input, and return different sequential information as output
- Intuition borrowed from **phrase-based machine translation**
  - Learn to convert one phrase of text into another

How do encoder-decoder models work?

- In NLP applications, encoders and decoders are often some type of **recurrent neural network**
- Encoders take sequential input and generate an **encoded representation** of it
  - This representation is undecipherable to casual observers!
- Decoders take this representation as input and generate a sequential (interpretable) output



Encoder-Decoder Chatbots

- It is important to incentivize response diversity to avoid repetitive (and boring) output
  - Mutual information objective function
  - Beam search
- Encoder-decoder chatbots also tend to struggle with modeling prior context and ensuring multi-turn coherence

What is the best way to evaluate chatbots?

- Best: **Human ratings**
- Automated metrics tend to correlate poorly with human judgement, especially when there are many and varied valid responses
  - Slot-filling accuracy
  - Word overlap with gold standard

Task-Based Dialogue Systems

- **Dialogue systems with a specific goal**
  - Generally, helping a user complete a task
- Most task-based dialogue systems are **frame-based**
  - Assume a set of user **intentions**
  - Each intention contains **slots** that can be filled by possible **values**
  - Related intentions or **frames** are sometimes called **domain ontologies**

Example Slots from a Travel Ontology

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?”

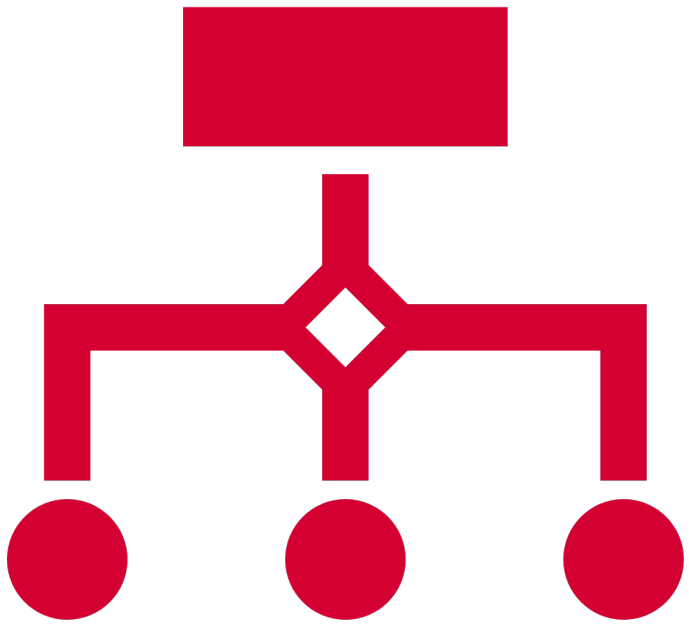
Control Structure for Frame-based Dialogue

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

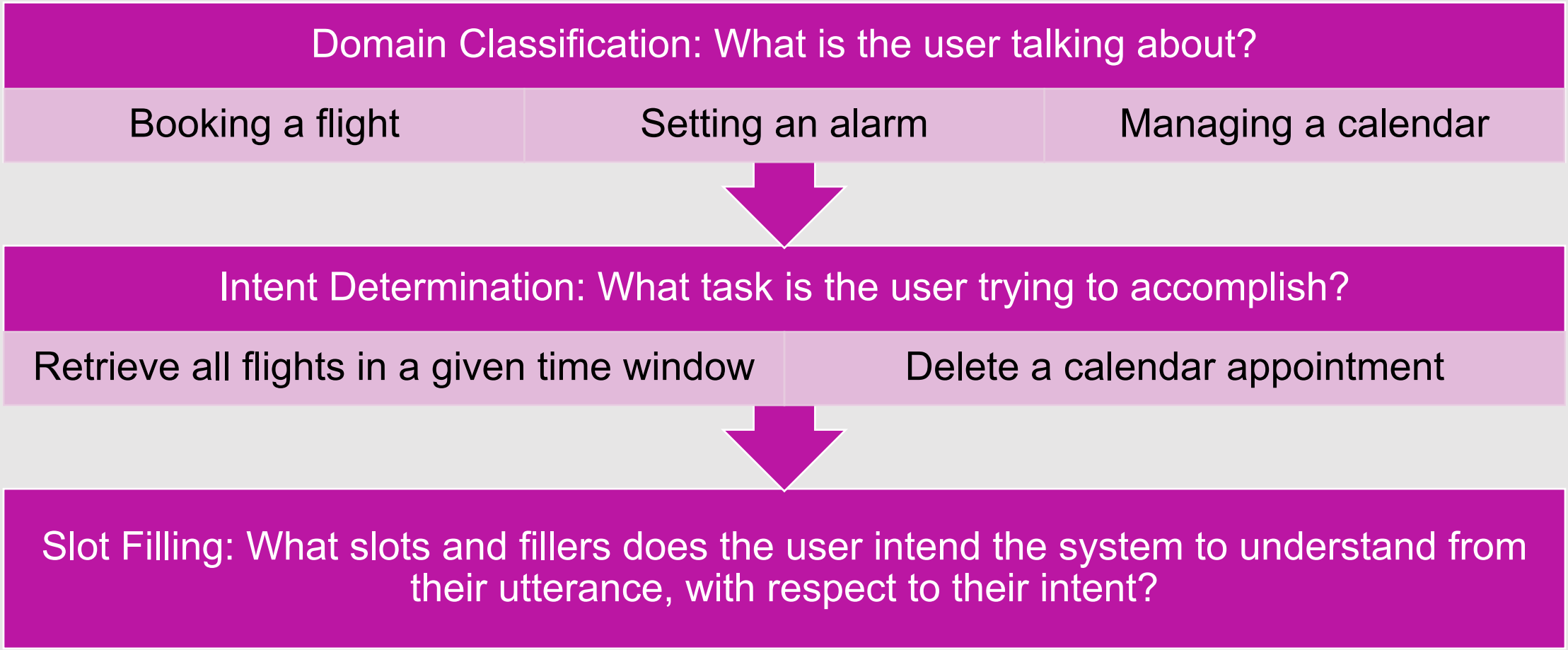
- Typically these questions are constructed using pre-specified question templates associated with each slot of each frame



Control Structure for Frame-based Dialogue

- The system continues questioning the user until it can fill all slots needed to perform the desired task
- 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

In a frame-based dialogue system, natural language understanding is necessary for performing three tasks:



How are slots usually filled?

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

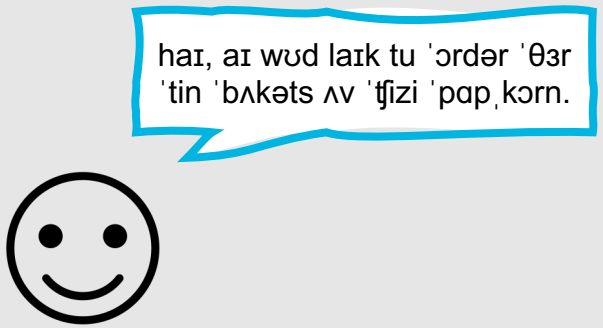
AMPM → am | pm

ORIGIN → from CITY

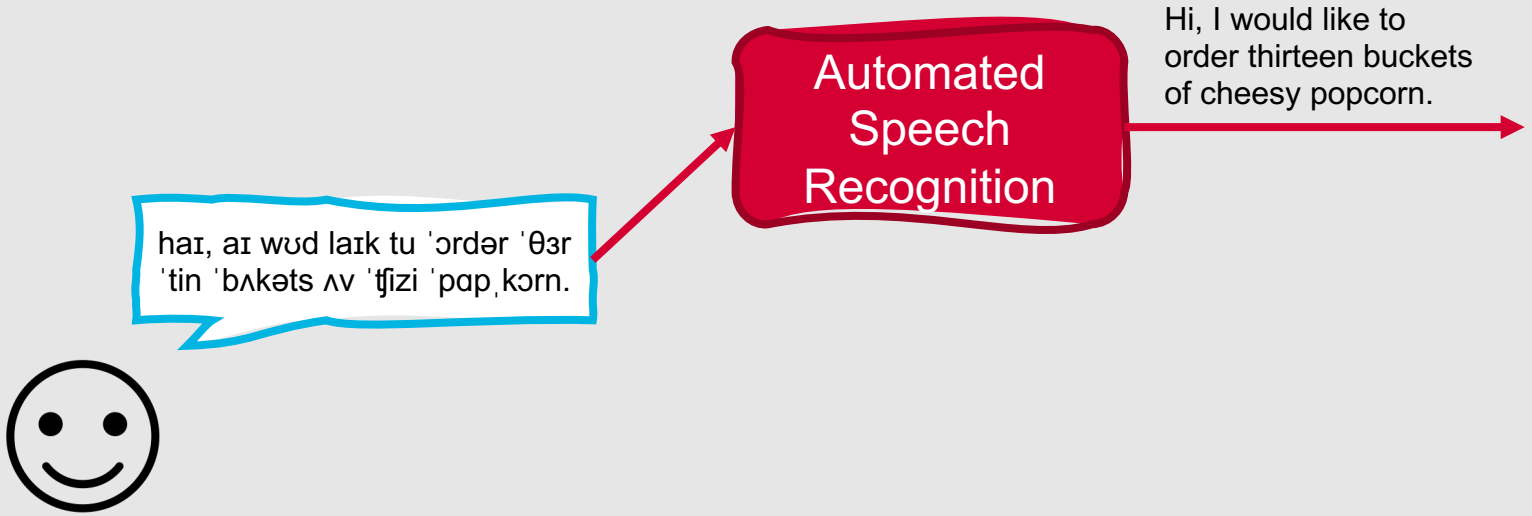
DESTINATION → to CITY

CITY → Chicago | Dallas | Denver | Phoenix

Dialogue State Architecture

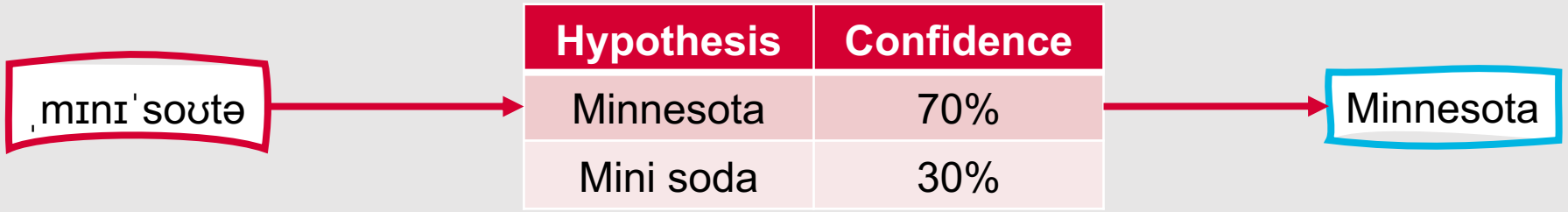


Dialogue State Architecture



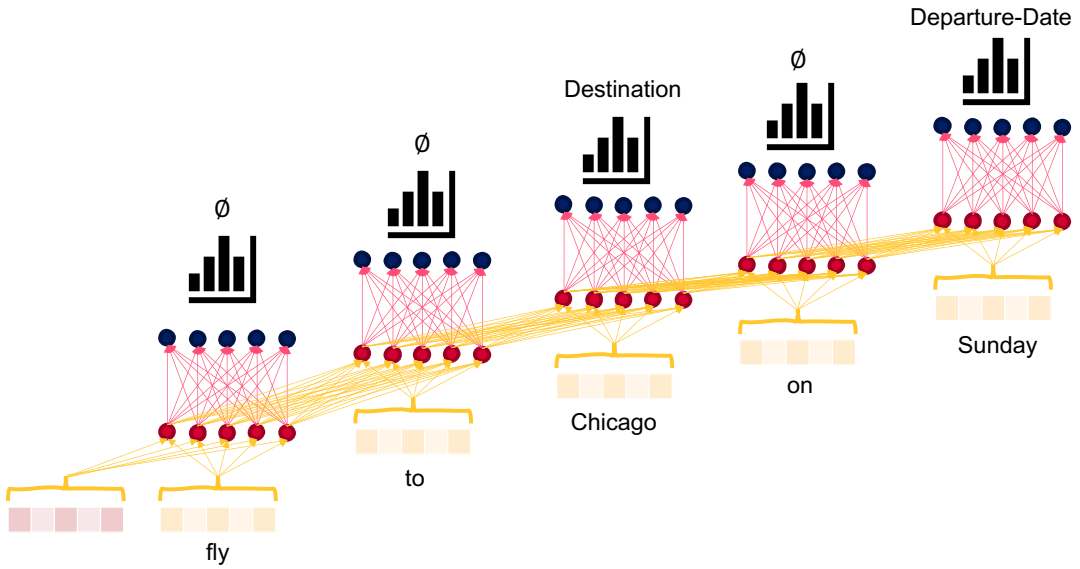
Automated Speech Recognition

- 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
  - System can use this score to determine whether to request clarification, or move forward on the assumption that the sequence is correct



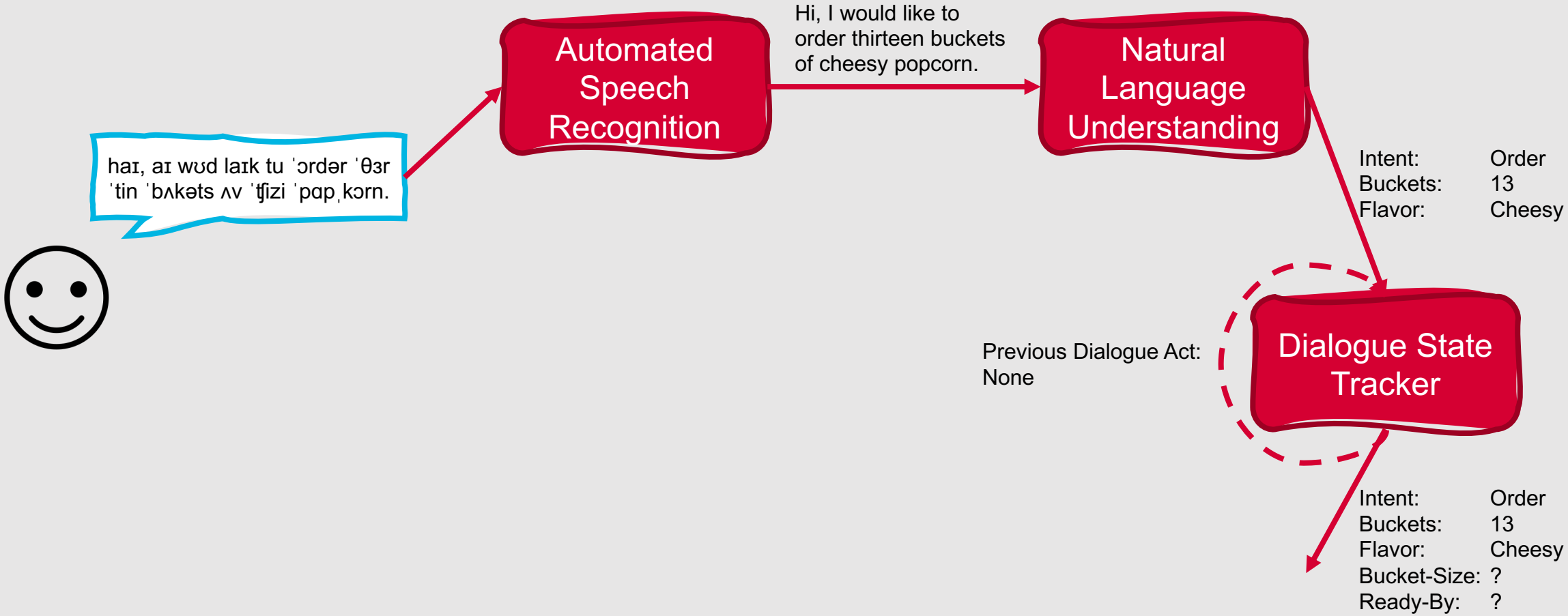
Dialogue State Architecture

Natural Language Understanding

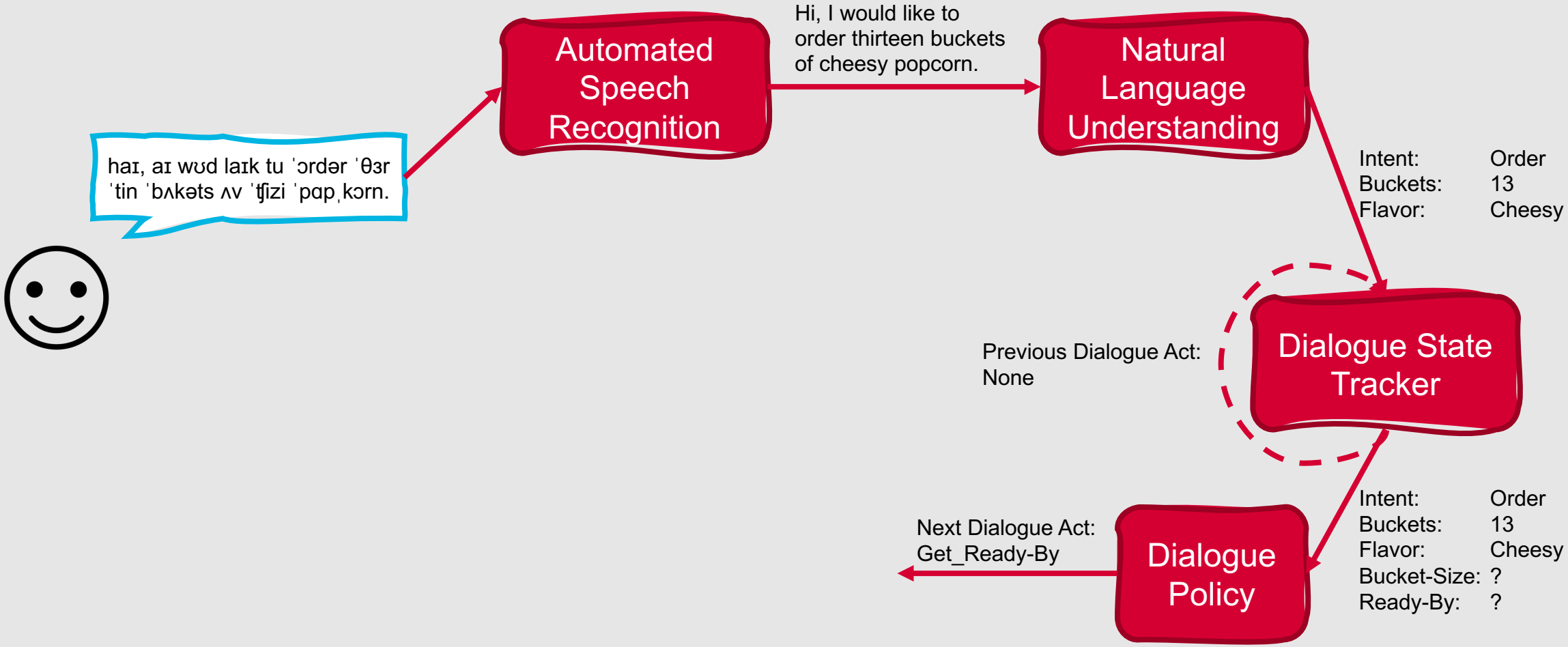


- Slot filling can be framed as a special case of supervised semantic parsing
  - Train a sequence model to map from input words to slot fillers, domain, and intent

Dialogue State Architecture



Dialogue State Architecture



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 a simple frame-based dialogue system, the system may just ask questions until the frame is 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

Sample Dialogue Act Tagset

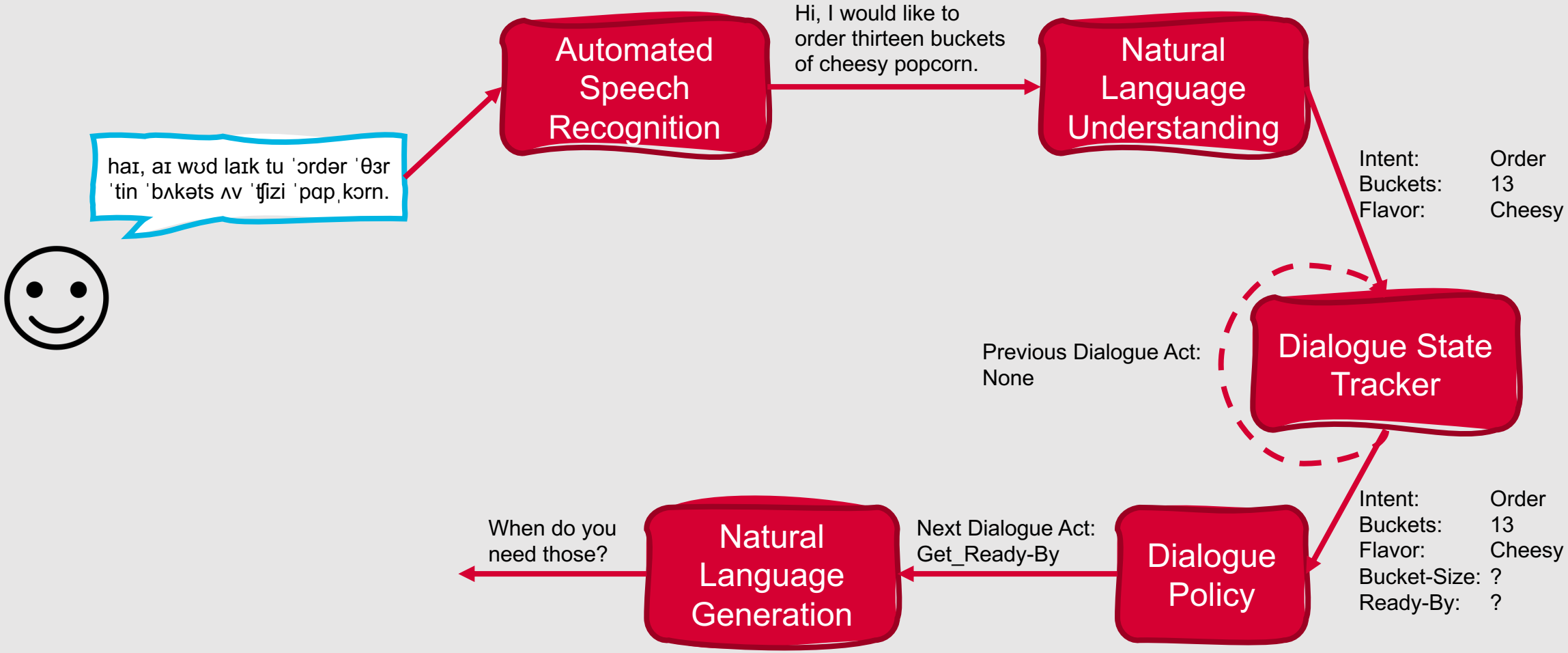
Tag	Valid System Act?	Valid User Act?	Description
Hello(a=x, b=y, …)	🙂	🙂	Open a dialogue and give info a=x, b=y, …
Inform(a=x, b=y, …)	🙂	🙂	Give info a=x, b=y, …
Request(a, b=x, …)	🙂	🙂	Request value for a given b=x, …
Reqalts(a=x, …)		🙂	Request alternative with a=x, …
Confirm(a=x, b=y, …)	🙂	🙂	Explicitly 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.	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 Petterino's is 123-456-7890	Inform(name="Petterino's", phone="123-456-7890")
U	Okay, thank you. Goodbye!	Bye()

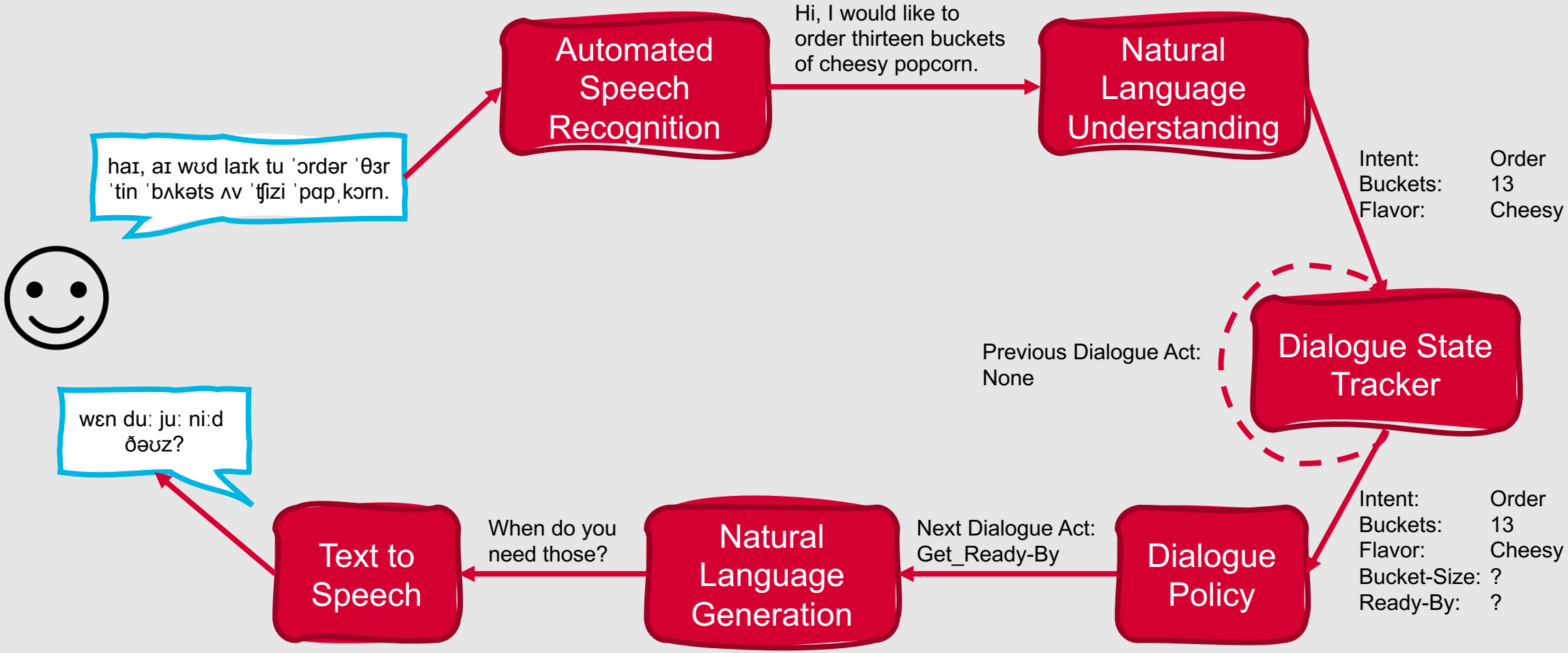
Dialogue State Architecture



Natural Language Generation

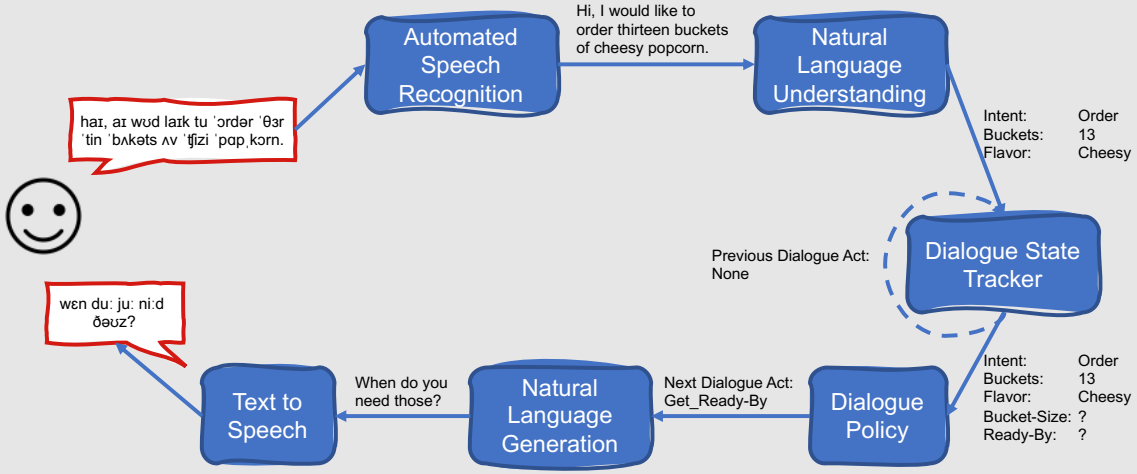
- In simpler systems, sentences are produced from pre-written templates
- In more sophisticated dialogue systems, the natural language generation component can be **conditioned on prior context** to produce more natural-sounding dialogue turns

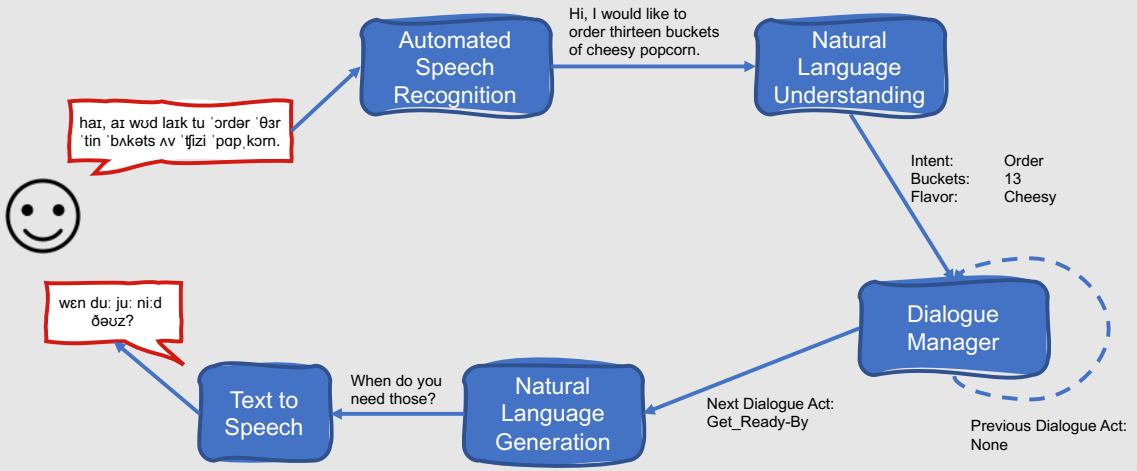
Dialogue State Architecture



Text to Speech Synthesis

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

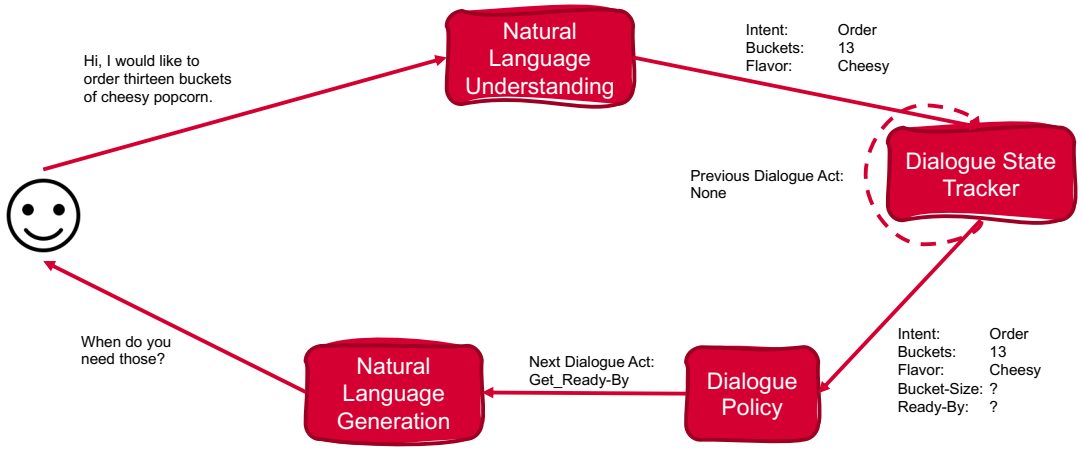




The dialogue state tracker and dialogue policy are sometimes grouped together as a single dialogue manager.

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



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

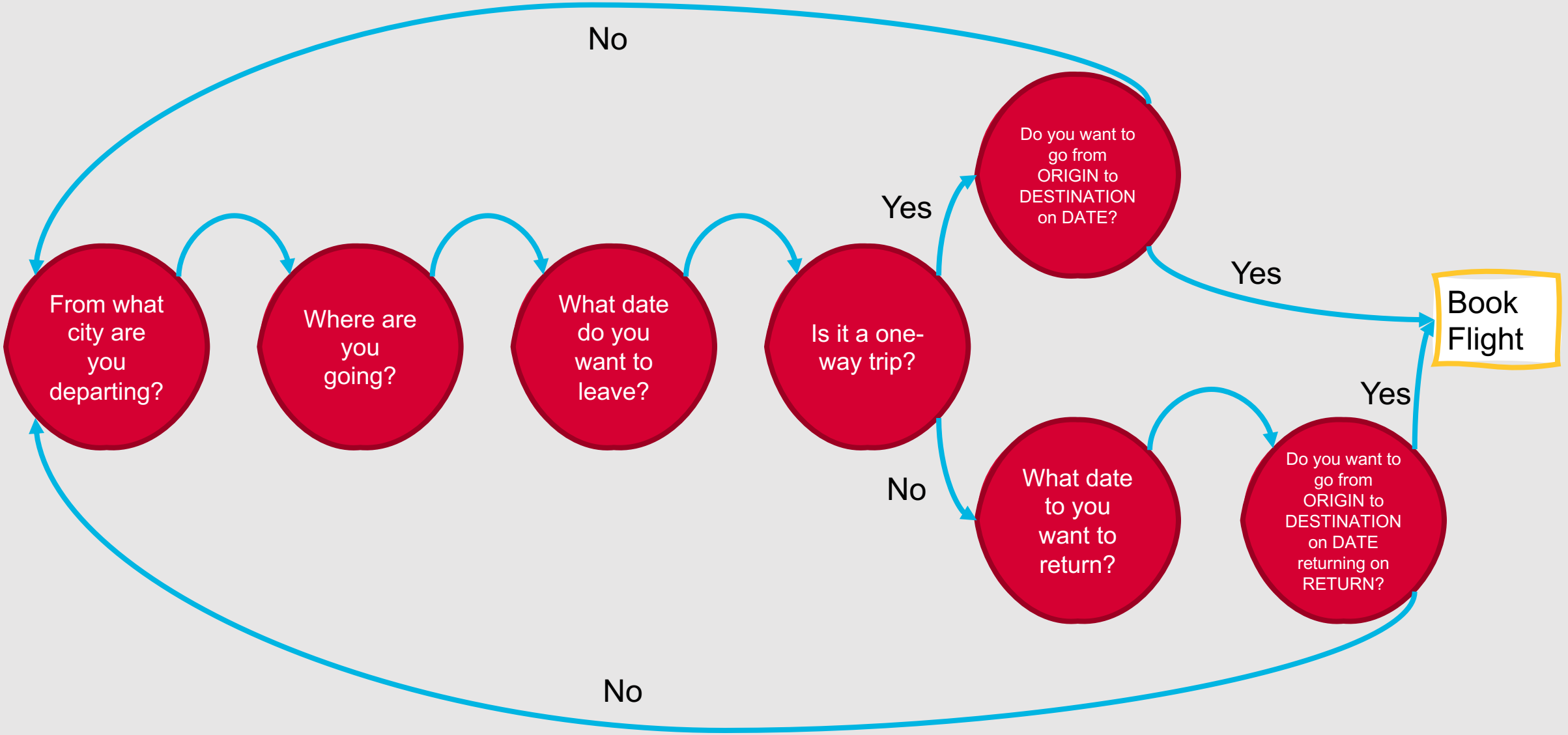
Transitions (arcs)

- Actions to take depending on how the user responds

System has full conversational initiative!

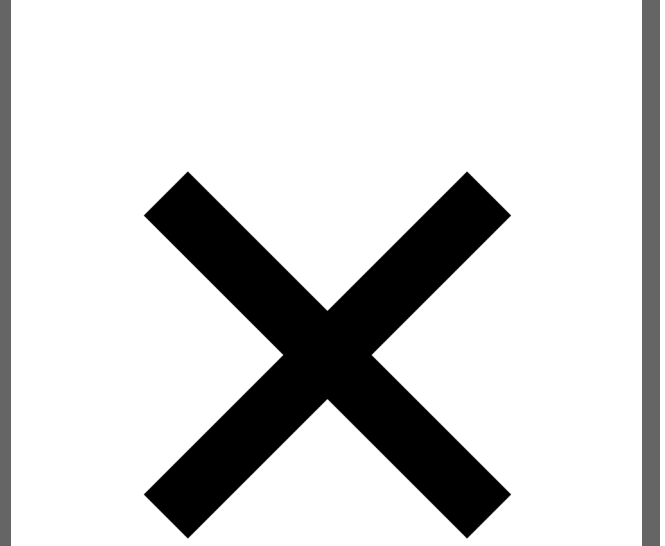
- Asks a series of questions
- Ignores or misinterprets inputs that are not direct answers to questions

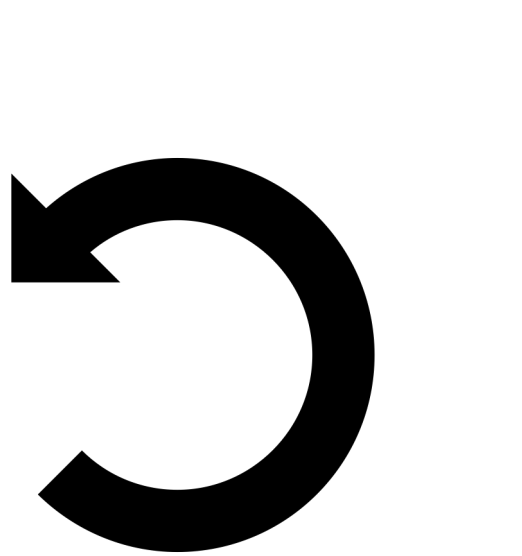
Finite State Dialogue Manager

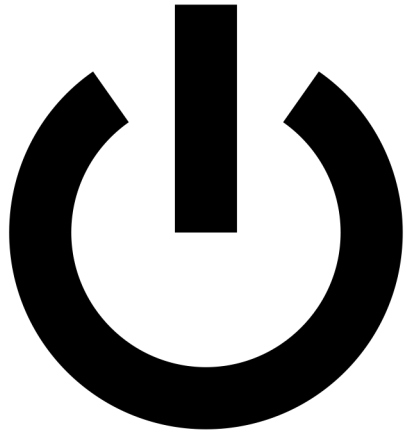


Finite State Dialogue Manager

- 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:

- Easy to implement
- Sufficient for simple tasks

Disadvantages:

- Can be awkward and annoying
- Cannot easily handle complex sentences

Other common dialogue management architectures make more complex use of dialogue state trackers and dialogue policy.

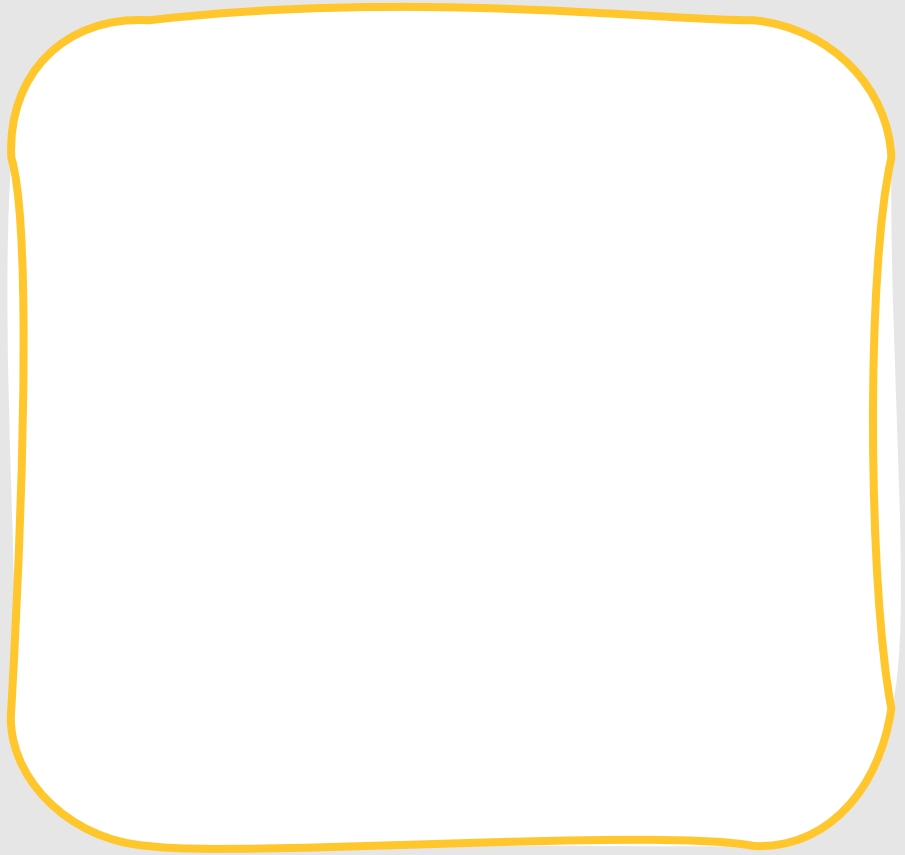
- Determine both:
  - The current state of the frame
    - What slots have been filled, and how, up to and including this point?
  - The user's most recent dialogue act

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)

Dialogue Policy

- Goal: Determine **what action the system should take next**
  - What dialogue act should be generated?
- Recently, this is often done using **neural networks** or **reinforcement learning**

NLG for Dialogue Systems

- Typically a two-stage process:
  - **Content Planning:** What should be said?
  - **Surface Realization:** How should it be said?

Content Planning

- Mostly handled by the dialogue policy
  - Selects a dialogue act
  - Selects which attributes to include in the dialogue act

Dialogue Act: Recommend

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

Surface Realization

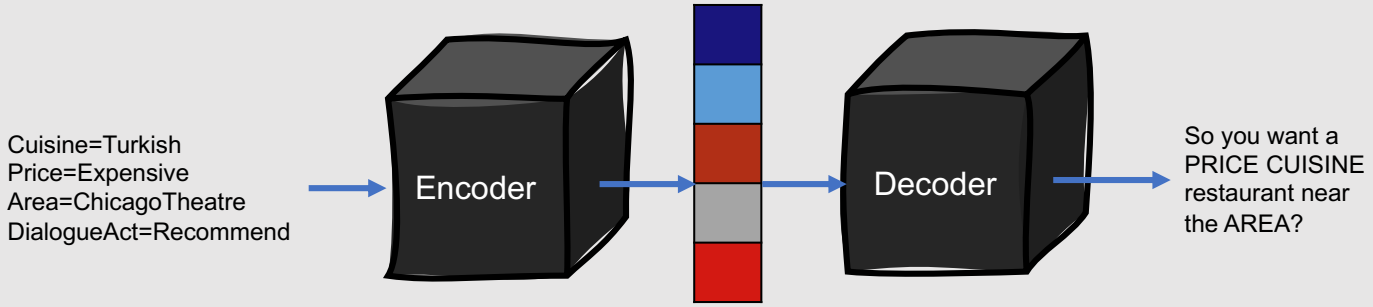
- Sentence of a specified type is generated, containing the specified attributes
- Often template-based
- Models can learn to generate templates using delexicalized training datasets
- Delexicalization: Replacing specific words in the training set that represent slot values with generic placeholder tokens

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.

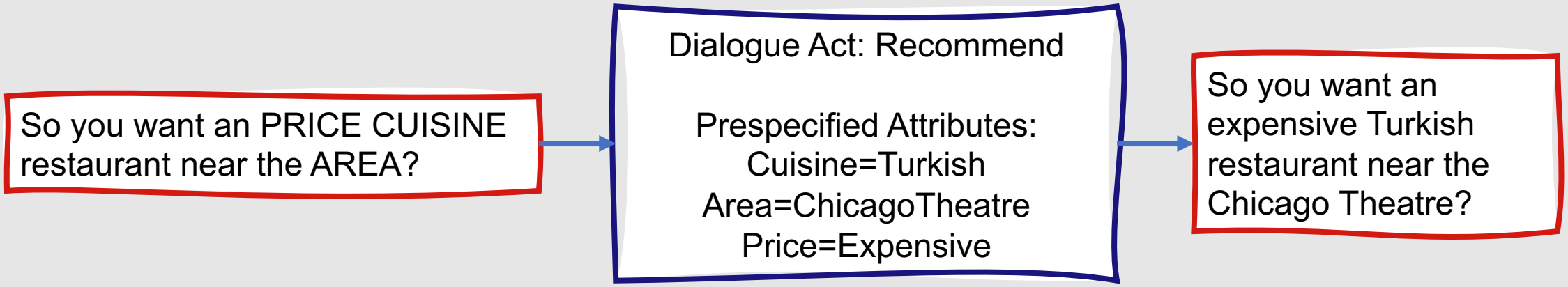
Mapping from Frames to Delexicalized Sentences



- 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

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



What about when systems (or users) make errors?

- 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

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

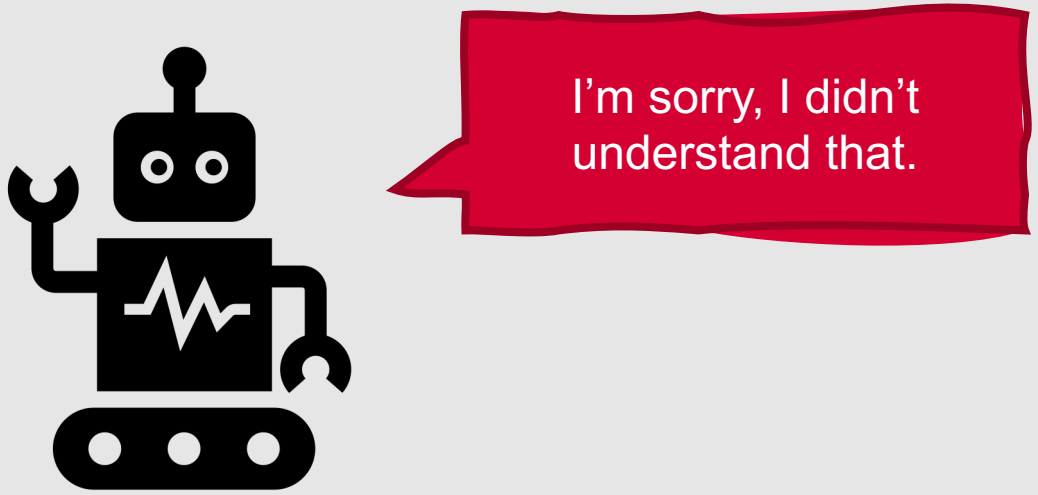
- Easier for users to correct the system's misrecognitions
  - User can just say "no" when the system tries to confirm
- But, can be awkward and unnatural
- Also extends the length of the conversation

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 my brother's party.

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 January 14th.

Other Strategies for Error Handling

- 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

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
  - $\text{Slot Error Rate} = \frac{\#\text{ of inserted, deleted, or substituted slots}}{\#\text{ of total reference slots}}$

What flights are going from Chicago to Dallas on the afternoon of August 31st?

Slot	Filler
ORIGIN	Chicago
DESTINATION	Denver
TIME	afternoon
DATE	8/31/2022

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

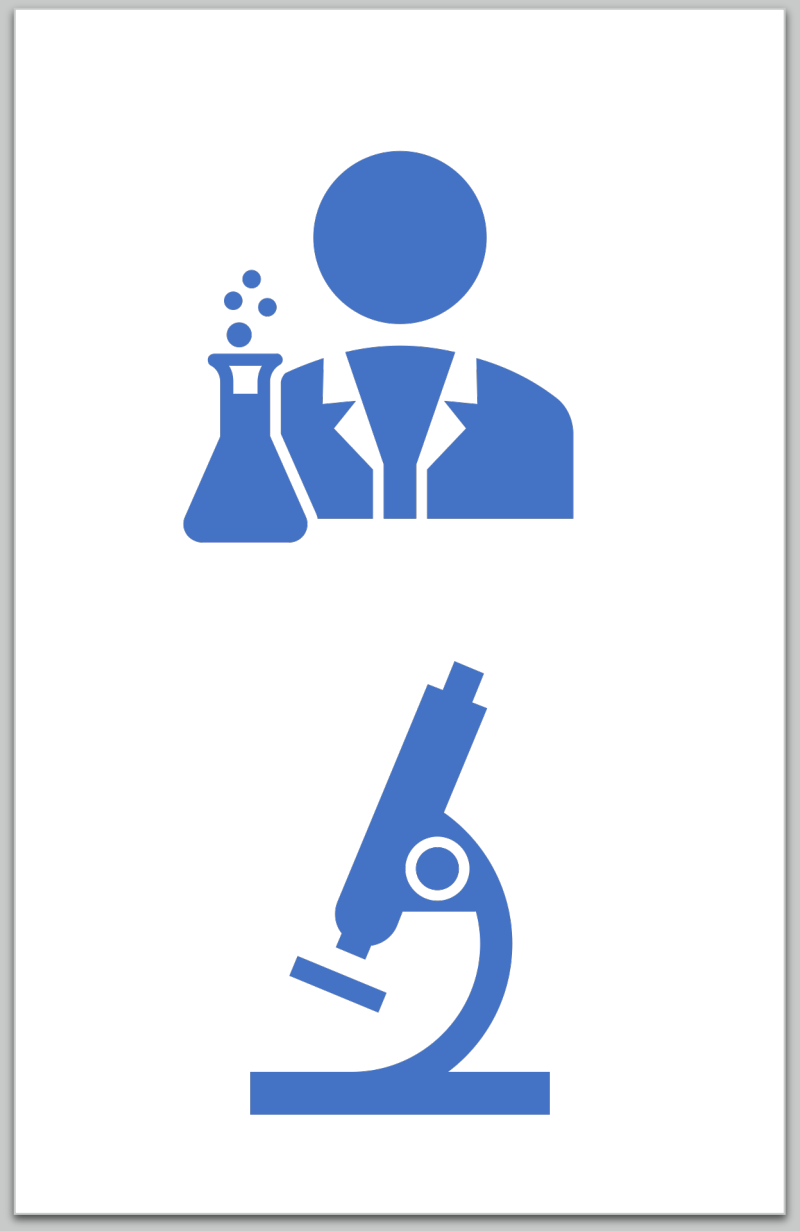
- How efficiently does the system help users perform tasks?
  - Total **elapsed time**
  - Number of **total turns**
  - Number of **system turns**
  - Number of **user queries**
  - **Turn correction ratio**
    - Number of system or user turns that were used solely to correct errors, divided by the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

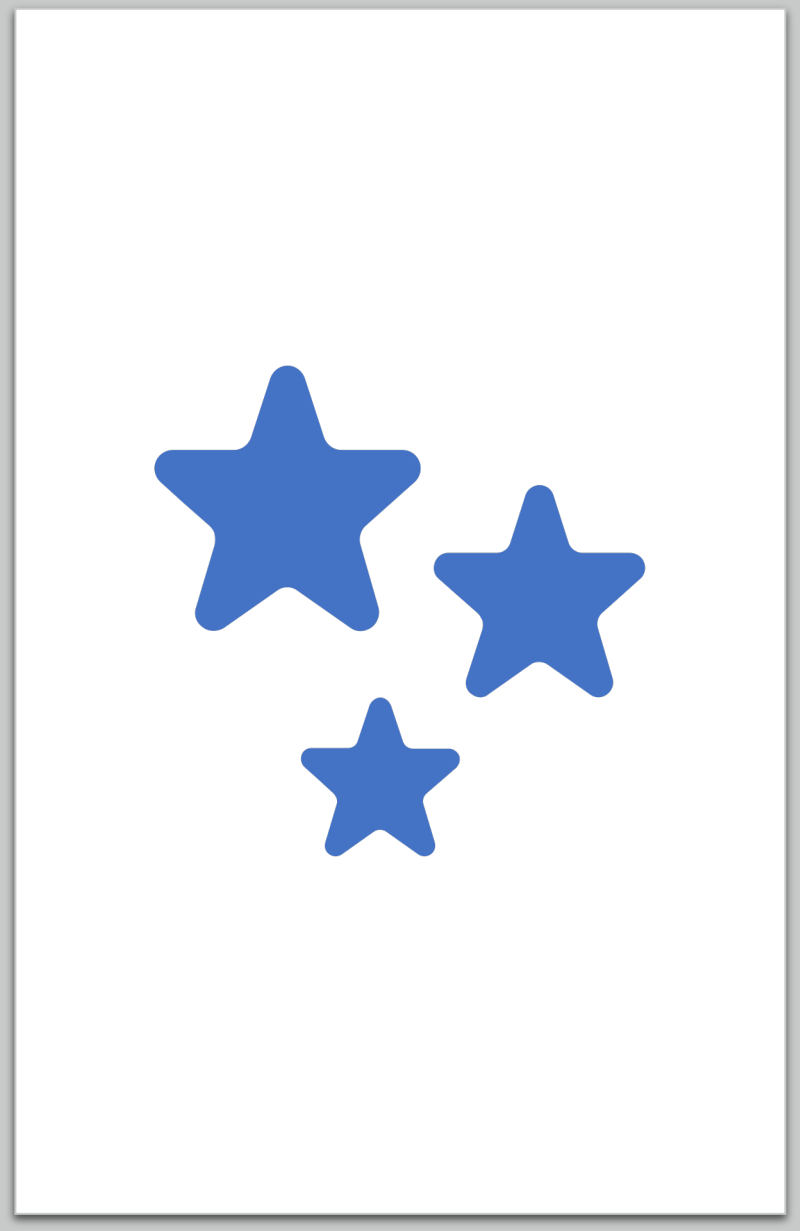
- Generally follows **user-centered design principles**
  1. Study the user and task
  2. Build simulations and prototypes
  3. Iteratively test the design on users

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
- 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
- Gender bias in conversational systems has recently been studied extensively:
  - Queens are Powerful too: Mitigating Gender Bias in Dialogue Generation: https://aclanthology.org/2020.emnlp-main.656.pdf
  - Does Gender Matter? Towards Fairness in Dialogue Systems: https://aclanthology.org/2020.coling-main.390.pdf
- Issues can also arise when chatbots are given problematic gender-conforming roles, or when they are designed to evade or respond politely to harassment

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!

Summary: Dialogue Systems and Chatbots

- Modern dialogue systems often contain mechanisms 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**