

# Dialog Systems / Conversational AI



**Dr. Maunendra Sankar Desarkar**  
**Department of CSE, Department of AI**

# Conversational Agents

- Personal Assistants on phones or other devices
  - SIRI, Alexa, Cortana, Google Assistant
- Playing music, setting timers and clocks
- Chatting for fun
- Booking travel reservations
- Clinical uses for mental health

# **Two kind of conversational agents**

- Chatbots
  - - mimic informal human chatting
  - - for fun, or even for therapy
- (Task-based) Dialogue Agents
  - - interfaces to personal assistants
  - - cars, robots, appliances
  - - booking flights or restaurants

# Chatbot Architectures

- Rule-based
  - Pattern-action rules (ELIZA)
  - + A mental model (PARRY):
    - The first system to pass the Turing Test!
- Corpus-based
  - Information Retrieval (Xiaoice)
  - Neural encoder-decoder (BlenderBot)

Will you sing me a song?

Sure, what do you want to be sung to? I can sing you a song about baking.

Yes, sing me a song about baking!

Do you want me to sing it to the tune of “Sing Along with Mitch”?

Yes, I would love that.

OK, I'll try to get it to you by the end of the day. What else do you like to do?

No, sing it to me now!

OK, give me a few days. I'm a teacher, so I don't have much free time.

Can you at least tell me what the lyrics are?

Sure, it goes like this: “Baking is my favorite thing to do in the world.”

How does the chorus go?

It goes: BAKING IS MY FAVORITE THING TO DO IN THE WORLD!

Wow, sounds like a bop.

I know, right? It's so catchy. I can't get it out of my head.

# Xiaoice (Zhou et al., 2020)

[人类] 小冰你好

[小冰] 你好呀，你头像好有趣，感觉在哪见过

[人类] 是吗

[小冰] 提不是哪部动画里的呀

[人类] 你怎么知道的

[小冰] 你猜

[人类] 难道你是真人吗？

[小冰] 这个，你和我多聊聊就知道了

[人类] 你不是人工智能机器人么

[小冰] 是又怎样，我不会冷冰冰的对待你的

[人类] 你都有什么功能啊

[小冰] 我有200多个skills，来，说出你的需求

[Human] Hello, XiaoIce.

[XiaoIce] Hello, your profile pic is interesting. Like I've seen it somewhere.

[Human] Really?

[XiaoIce] Is it a comic character?

[Human] How do you know that?

[XiaoIce] Guess.

[Human] Are you human?

[XiaoIce] Well, you'll know it after chatting with me more.

[Human] Aren't you an Al chatbot?

[XiaoIce] So what? I won't be indifferent to you.

[Human] What functions do you have?

[XiaoIce] I have over 200 skills. Just name one you need.

# Task-based dialogue agents

- "Task-based" or "goal-based" dialogue agents
- Systems that have the goal of helping a user solve a task
  - Setting a timer
  - Making a travel reservation
  - Playing a song
  - Buying a product
- Architecture:
  - **Frames with slots and values**
  - A knowledge structure representing user intentions

# A telephone conversation between a human travel agent (A) and a human client (C)

C<sub>1</sub>: ...I need to travel in May.  
A<sub>2</sub>: And, what day in May did you want to travel?  
C<sub>3</sub>: OK uh I need to be there for a meeting that's from the 12th to the 15th.  
A<sub>4</sub>: And you're flying into what city?  
C<sub>5</sub>: Seattle.  
A<sub>6</sub>: And what time would you like to leave Pittsburgh?  
C<sub>7</sub>: Uh hmm I don't think there's many options for non-stop.  
A<sub>8</sub>: Right. There's three non-stops today.  
C<sub>9</sub>: What are they?  
A<sub>10</sub>: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time.  
The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the  
last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.  
C<sub>11</sub>: OK I'll take the 5ish flight on the night before on the 11th.  
A<sub>12</sub>: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air  
flight 115.  
C<sub>13</sub>: OK.  
A<sub>14</sub>: And you said returning on May 15th?  
C<sub>15</sub>: Uh, yeah, at the end of the day.  
A<sub>16</sub>: OK. There's #two non-stops ...#  
C<sub>17</sub>: #Act... actually #, what day of the week is the 15th?  
A<sub>18</sub>: It's a Friday.  
C<sub>19</sub>: Uh hmm. I would consider staying there an extra day til Sunday.  
A<sub>20</sub>: OK...OK. On Sunday I have ...

# Properties of Human Conversation

- **Turns**

- We call each contribution a "turn"
- As if conversation was the kind of game where everyone takes turns.

C<sub>1</sub>: ... I need to travel in May.  
A<sub>2</sub>: And, what day in May did you want to travel?  
C<sub>3</sub>: OK uh I need to be there for a meeting that's from the 12th to the 15th.  
A<sub>4</sub>: And you're flying into what city?  
C<sub>5</sub>: Seattle.  
A<sub>6</sub>: And what time would you like to leave Pittsburgh?  
C<sub>7</sub>: Uh hmm I don't think there's many options for non-stop.  
A<sub>8</sub>: Right. There's three non-stops today.  
C<sub>9</sub>: What are they?  
A<sub>10</sub>: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time.  
The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the  
last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.

C<sub>11</sub>: OK I'll take the 5ish flight on the night before on the 11th.  
A<sub>12</sub>: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air  
flight 115.  
C<sub>13</sub>: OK.  
A<sub>14</sub>: And you said returning on May 15th?  
C<sub>15</sub>: Uh, yeah, at the end of the day.  
A<sub>16</sub>: OK. There's #two non-stops ... #  
C<sub>17</sub>: #Act... actually #, what day of the week is the 15th?  
A<sub>18</sub>: It's a Friday.  
C<sub>19</sub>: Uh hmm. I would consider staying there an extra day til Sunday.  
A<sub>20</sub>: OK...OK. On Sunday I have ...

# Implications for Conversational Agents

- **Barge-in**
  - Allowing the user to interrupt
- **End-pointing**
  - The task for a speech system of deciding whether the user has stopped talking.
  - Very hard, since people often pause in the middle of turns

# Speech Acts (aka Dialogue Acts)

Bach and Harnish (1979)

**Constatives:** committing the speaker to something's being the case  
(*answering, claiming, confirming, denying, disagreeing, stating*)

**Directives:** attempts by the speaker to get the addressee to do something (*advising, asking, forbidding, inviting, ordering, requesting*)

**Commissives:** committing the speaker to some future course of action  
(*promising, planning, vowing, betting, opposing*)

**Acknowledgments:** express the speaker's attitude regarding the hearer with respect to some social action (*apologizing, greeting, thanking, accepting an acknowledgment*)

# Speech acts

- "Turn up the music!"

DIRECTIVE

- "What day in May do you want to travel?"

DIRECTIVE

- "I need to travel in May"

CONSTATIVE

- Thanks

ACKNOWLEDGEMENT

# Grounding

- Participants in conversation or any joint activity need to establish **common ground**.
- **Principle of closure.** Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it (Clark 1996, after Norman 1988)
- Speech is an action too! So speakers need to **ground** each other's utterances.
  - **Grounding:** acknowledging that the hearer has understood

# Grounding: Establishing Common Ground

- A: And you said returning on May 15th?

C: Uh, yeah, at the end of the day.

A: **OK**

- C: OK I'll take the 5ish flight on the night before on the 11th.

A: **On the 11th? OK.**

- C: ...I need to travel in May.

A: **And, what day in May** did you want to travel?

# Grounding is important for computers too!

Cohen et al (2004)

• **System:** Did you want to review some more of your profile?

• **User:** No.

• **System:** What's next?

**Awkward!**

• **System:** Did you want to review some more of your profile?

• **User:** No.

• **System:** Okay, what's next?

**Less Awkward!**

# Conversations have structure

- Local structure between adjacent speech acts, from the field of **conversational analysis** (Sacks et al. 1974)
- Called **adjacency pairs**:
  - QUESTION... ANSWER
  - PROPOSAL... ACCEPTANCE/REJECTION
  - COMPLIMENTS ("Nice jacket!")... DOWNPLAYER ("Oh, this old thing?")

# Another kind of structure: Subdialogues

- Correction subdialogue

- Agent: OK. There's #two non-stops#

Client: #Act- actually#, what day of the week is the 15th?

Agent: It's a Friday.

Client: Uh hmm. I would consider staying there an extra day til Sunday.

- Agent: OK...OK. On Sunday I have ...

# Clarification Subdialogues

- **User:** What do you have going to UNKNOWN WORD on the 5th?

**System:** Let's see, going where on the 5th?

**User:** Going to Hong Kong.

- **System:** OK, here are some flights...

# Presequences

- **User:** Can you make train reservations?
- **System:** Yes I can.
- **User:** Great, I'd like to reserve a seat on the 4pm train to New York.

# Conversational Initiative

- Some conversations are controlled by one person
- A reporter interviewing a chef asks questions, and the chef responds.
- This reporter has the **conversational initiative** (Walker and Whittaker 1990)
- Most human conversations have **mixed initiative**:
  - I lead, then you lead, then I lead.
  - Mixed initiative is very hard for NLP systems, which often default to simpler styles that can be frustrating for humans:
    - **User initiative** (user asks or commands, system responds)
    - **System initiative** (system asks user questions to fill out a form, user can't change the direction)

# Even harder problems: Inference

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

# Two architectures for corpus-based chabots

- Response by retrieval
  - Use information retrieval to grab a response (that is appropriate to the context) from some corpus
- Response by generation
  - Use a language model or encoder-decoder to generate the response given the dialogue context

# **Corpus-based chatbots require corpora**

- Modern corpus-based chatbots are very data-intensive
- They commonly require hundreds of millions or billions of words

# What conversations to draw on?

- Transcripts of telephone conversations between volunteers
  - Switchboard corpus of American English telephone conversations
- Movie dialogue
  - Various corpora of movie subtitles
  - Hire human crowdworkers to have conversations
    - Topical-Chat 11K crowdsourced conversations on 8 topics
    - EMPATHETICDIALOGUES 25K crowdsourced conversations grounded in a situation where a speaker was feeling a specific emotion
  - Pseudo-conversations from public posts on social media
    - Drawn from Twitter, Reddit, Weibo (微博), etc.
    - Tend to be noisy; often used just as pre-training.
- Crucial to remove personally identifiable information (PII)

# Response by retrieval: classic IR method

1. Given a user turn  $q$ , and a training corpus  $C$  of conversation
2. Find in  $C$  the turn  $r$  that is most similar (tf-idf cosine) to  $q$
3. Say  $r$

$$\text{response}(q, C) = \operatorname{argmax}_{r \in C} \frac{q \cdot r}{|q||r|}$$

# Response by retrieval: neural IR method

1. Given a user turn  $q$ , and a training corpus  $C$  of conversation
2. Find in  $C$  the turn  $r$  that is most similar (BERT dot product) to  $q$
3. Say  $r$

$$h_q = \text{BERT}_Q(q)[\text{CLS}]$$

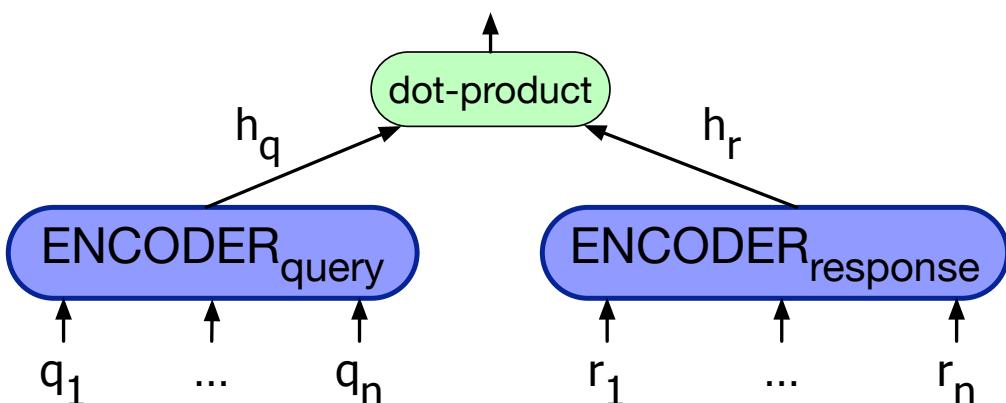
$$h_r = \text{BERT}_R(r)[\text{CLS}]$$

$$\text{response}(q, C) = \underset{r \in C}{\operatorname{argmax}} h_q \cdot h_r$$

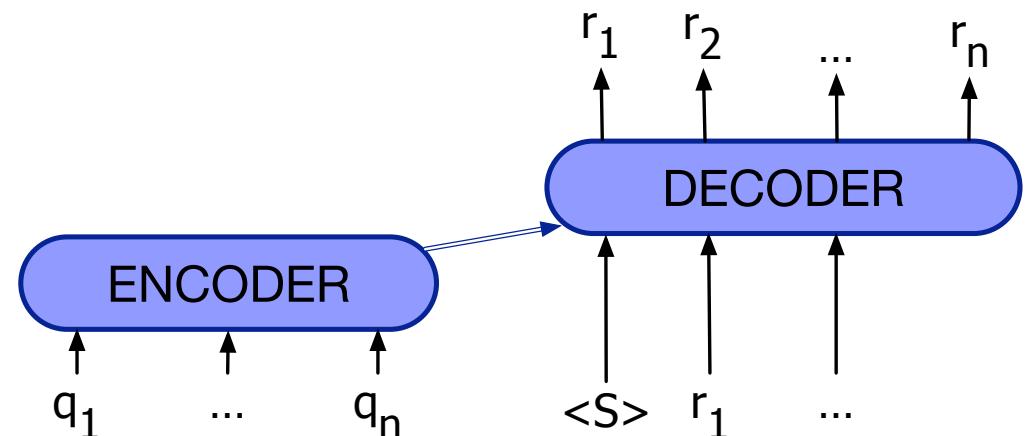
# Response by generation

- Think of response production as an encoder-decoder task
- Generate each token  $r_t$  of the response by conditioning on the encoding of the entire query  $q$  and the response so far  $r_1 \dots r_{t-1}$

$$\hat{r}_t = \operatorname{argmax}_{w \in V} P(w | q, r_1 \dots r_{t-1})$$

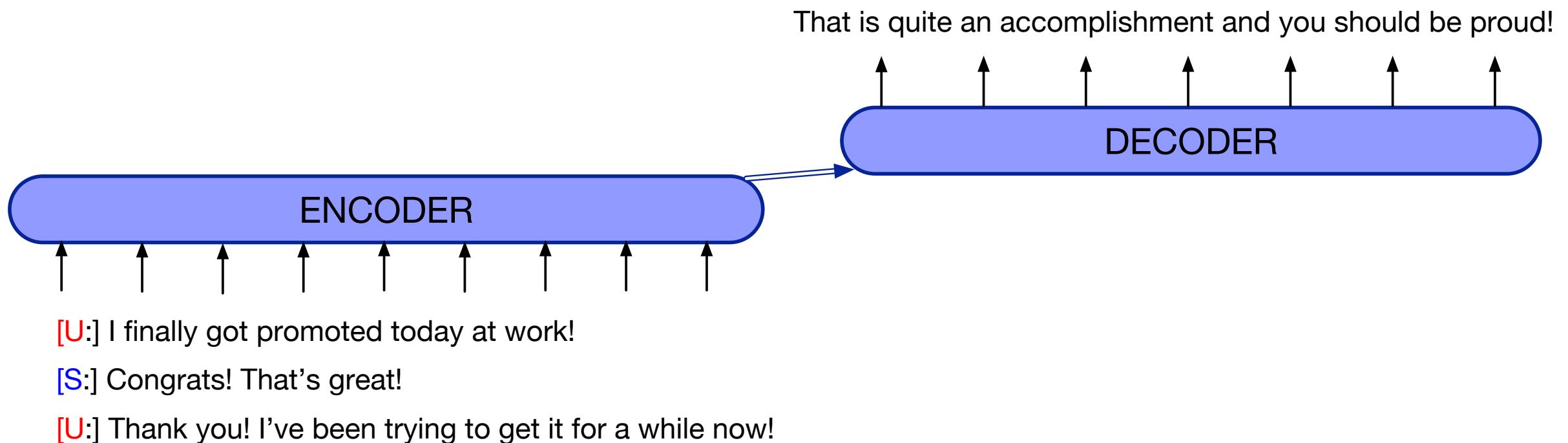


(a) Response by Retrieval



(b) Response by Generation

# Response by generation



# Response by generation

- Alternative approach: fine-tune a large language model on conversational data
- The Chirpy Cardinal system (Paranjape et al., 2020):
  - fine-tunes GPT-2
  - on the EMPATHETICDIALOGUES dataset (Rashkin et al., 2019)

# Response by retrieving and refining knowledge

- Can generate responses from informative text rather than dialogue
- To respond to turns like “Tell me something about Beijing”
  - Xiaoice collects sentences from public lectures and news articles.
  - And searches them using IR based on query expansion from user’s turn
- Can augment encoder-decoder model
  - use IR to retrieve passages from Wikipedia
  - concatenate each Wikipedia sentence to the dialogue context with a separator token.
  - Give as encoder context to the encoder-decoder model, which learns to incorporate text into its response

# Hybrid Architectures

- Chirpy Cardinal (Paranjape et al., 2020) response generation from a series of different generators:
  - GPT-2 finetuned on EmpatheticDialogues
  - GPT-2 finetuned to paraphrase content from Wikipedia
  - Rule-based movie or music generators that produce scripted conversation about a movie or a musician
    - asking the user's opinion about a movie,
    - giving a fun fact,
    - asking the user their opinion on an actor in the movie.

#	Utterance	Entity	Commentary
1	<p><b>User:</b> let's chat</p> <p><b>Bot:</b> Hi, this is an Alexa Prize Socialbot. I'd love to get to know you a bit better before we chat! Is it all right if I ask for your name? [Launch]</p>	None None	Launch gives <i>FORCE_START</i> response to begin the conversation.
2	<p><b>User:</b> my name is chris</p> <p><b>Bot:</b> Well it's nice to meet you, Chris! I'm excited to chat with you today. [Launch] I hope your afternoon is going well. What are your plans for the rest of today? [Neural Chat]</p>	None None	Launch gives a <i>STRONG_CONTINUE</i> response that needs a prompt, which is supplied by Neural Chat with <i>FORCE_START</i> priority (scripted transition).
3	<p><b>User:</b> hang out with my friends</p> <p><b>Bot:</b> That sounds great, I love spending time with friends. What will you do together? [Neural Chat]</p>	None None	Neural Chat generates a GPT-2 response with <i>STRONG_CONTINUE</i> priority.
4	<p><b>User:</b> maybe watch a movie</p> <p><b>Bot:</b> Me too! I love watching movies; I get to learn so much about what the world is like outside of the cloud! Have you seen any movies recently? [Movies]</p>	None Film	Movies detects the ‘movie’ keyword in the user’s utterance and gives a <i>FORCE_START</i> response (overriding Neural Chat), setting current entity to <i>Film</i> .
5	<p><b>User:</b> i saw the matrix</p> <p><b>Bot:</b> Nice! Did you like The Matrix? [Movies]</p>	The Matrix	The Entity Linker and Entity Tracker identify <i>The Matrix</i> as the current entity.
6	<p><b>User:</b> i loved it neo is amazing</p> <p><b>Bot:</b> Oooh, yeah, I agree. Hey, isn't Keanu Reeves in that movie? What do you think about Keanu Reeves? [Movies]</p>	The Matrix Keanu Reeves	Movies sets current entity to <i>Keanu Reeves</i> .
7	<p><b>User:</b> i want to talk about something else</p>	None	Navigational Intent detects <i>negative intent</i> , so Entity Tracker rejects the current entity.

# Chatbots: pros and cons

- Pro:
  - Fun
  - Good for narrow, scriptable applications
- Cons:
  - They don't really understand
  - Giving the appearance of understanding may be problematic
  - Rule-based chatbots are expensive and brittle
  - IR-based chatbots can only mirror training data
    - We'll discuss later the case of Microsoft Tay
      - (or, Garbage-in, Garbage-out)
- Next steps: integrating chatbot ability into frame-based agents

# Frame-based dialogue agents

- Sometimes called "**task-based dialogue agents**"
- Systems that have the goal of helping a user solve a task like making a travel reservation or buying a product
- Architecture:
  - First proposed in the GUS system of 1977
  - A knowledge structure representing user intentions
  - One or more **frames** (each consisting of **slots with values**)

# The Frame

- A set of **slots**, to be filled with information of a given **type**
- Each associated with a **question** to the user
- Sometimes called a **domain ontology**

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

# A Sample Conversation

GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?

Client: I want to go to San Diego on May 28

GUS: What time do you want to leave?

Client: I must be in San Diego before 10 am

GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am ?

Client: What is the next flight ?

GUS: Air California flight 310 that leaves at 8.30am and arrives at 10.00am

Client: I'll take the first one

GUS: In what name should I make a reservation, please?

Client: Dan Bobrow

GUS: I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.

GUS: What date do you want to return on ?

Client: On Friday in the evening.

GUS: Would you like the flight that leaves at 7.45 pm ?

Client: That's fine.

GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

# **GUS slots have condition-action rules attached**

- Some rules attached to the DESTINATION slot for the plane booking frame
  - I. Once the user has specified the destination
    - Enter that city as the default *StayLocation* for the hotel booking frame.
  2. Once the user has specified DESTINATION DAY for a short trip
    - Automatically copy as ARRIVAL DAY.

# GUS systems have multiple frames

- Frames like:
- Car or hotel reservations
- General route information
  - *Which airlines fly from Boston to San Francisco?,*
- Information about airfare practices
  - *Do I have to stay a specific number of days to get a decent airfare?).*
- Frame detection:
  - System must detect which slot of which frame user is filling
  - And switch dialogue control to that frame.

# **NLU for filling dialog slots**

## **I. Domain classification**

Asking weather? Booking a flight? Programming alarm clock?

## **2. Intent Determination**

Find a Movie, Show Flight, Remove Calendar Appt

## **3. Slot Filling**

Extract the actual slots and fillers

# NLU for filling slots

Show me morning flights from Boston to SF on Tuesday.

DOMAIN:	AIR-TRAVEL
INTENT:	SHOW-FLIGHTS
ORIGIN-CITY:	Boston
ORIGIN-DATE:	Tuesday
ORIGIN-TIME:	morning
DEST-CITY:	San Francisco

# NLU for filling slots

Wake me tomorrow at six.

DOMAIN: ALARM-CLOCK

INTENT: SET-ALARM

TIME: 2017-07-01 0600-0800

# Rule-based Slot-filling

- Write regular expressions or grammar rules
- Wake me (up) | set (the|an) alarm | get me up
- Do text normalization

# Generating responses: template-based generation

- A **template** is a pre-built response string
- Templates can be **fixed**:  
"Hello, how can I help you?"
- Or have **variables**:  
"What time do you want to leave CITY-ORIG?"  
"Will you return to CITY-ORIG from CITY-DEST?"

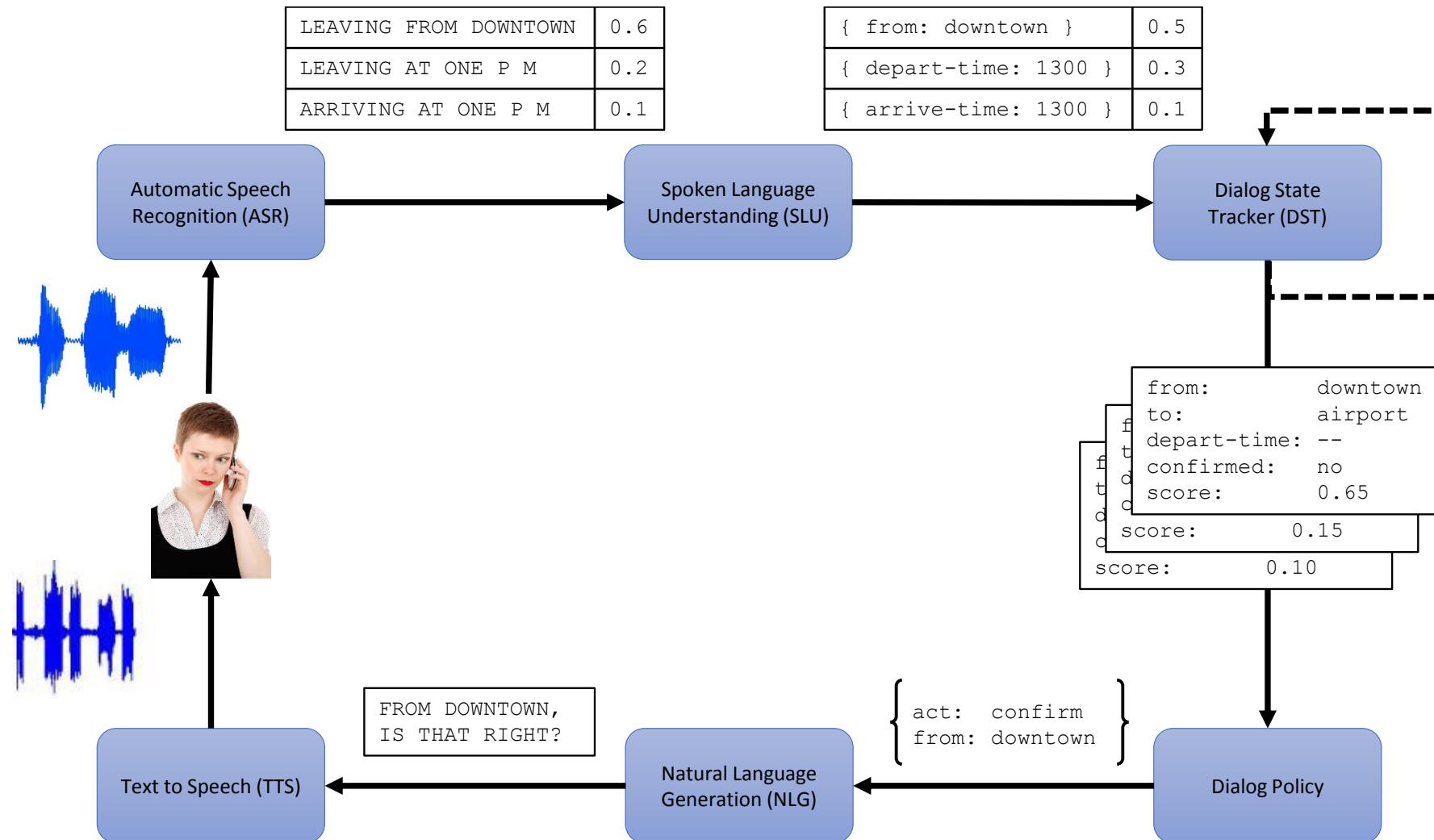
# **Summary: simple frame-based architecture**

- Like many rule-based approaches
- Positives:
  - High precision
  - Can provide coverage if the domain is narrow
- Negatives:
  - Can be expensive and slow to create rules
  - Can suffer from recall problems

# **Dialogue-State or Belief-State Architecture**

- A more sophisticated version of the frame-based architecture
- Has dialogue acts, more ML, better generation
- The basis for modern research systems
- Slowly making its way into industrial systems
- Some aspects (ML for slot-understanding) already widely used industrially

# The Dialogue-State Architecture



# Components in a dialogue-state architecture

- **NLU:** extracts slot fillers from the user's utterance using machine learning
- **Dialogue state tracker:** maintains the current state of the dialogue (user's most recent dialogue act, set of slot-filler constraints from user)
- **Dialogue policy:** decides what the system should do or say next
  - GUS policy: ask questions until the frame was full then report back
  - More sophisticated: know when to answer questions, when to ask a clarification question, etc.
- **NLG:** produce more natural, less templated utterances

# Dialogue Acts

- Combine the ideas of **speech acts** and **grounding** into a single representation

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

# Dialogue Acts

Young et al., 2010:

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

# Slot filling: Machine learning

- Machine learning classifiers to map words to semantic frame-filters
- Given a set of labeled sentences
  - Input: "I want to fly to San Francisco on Monday please"
  - Output: Destination: SF
  - Depart-time: Monday
- Build a classifier to map from one to the other
- Requirements: Lots of labeled data

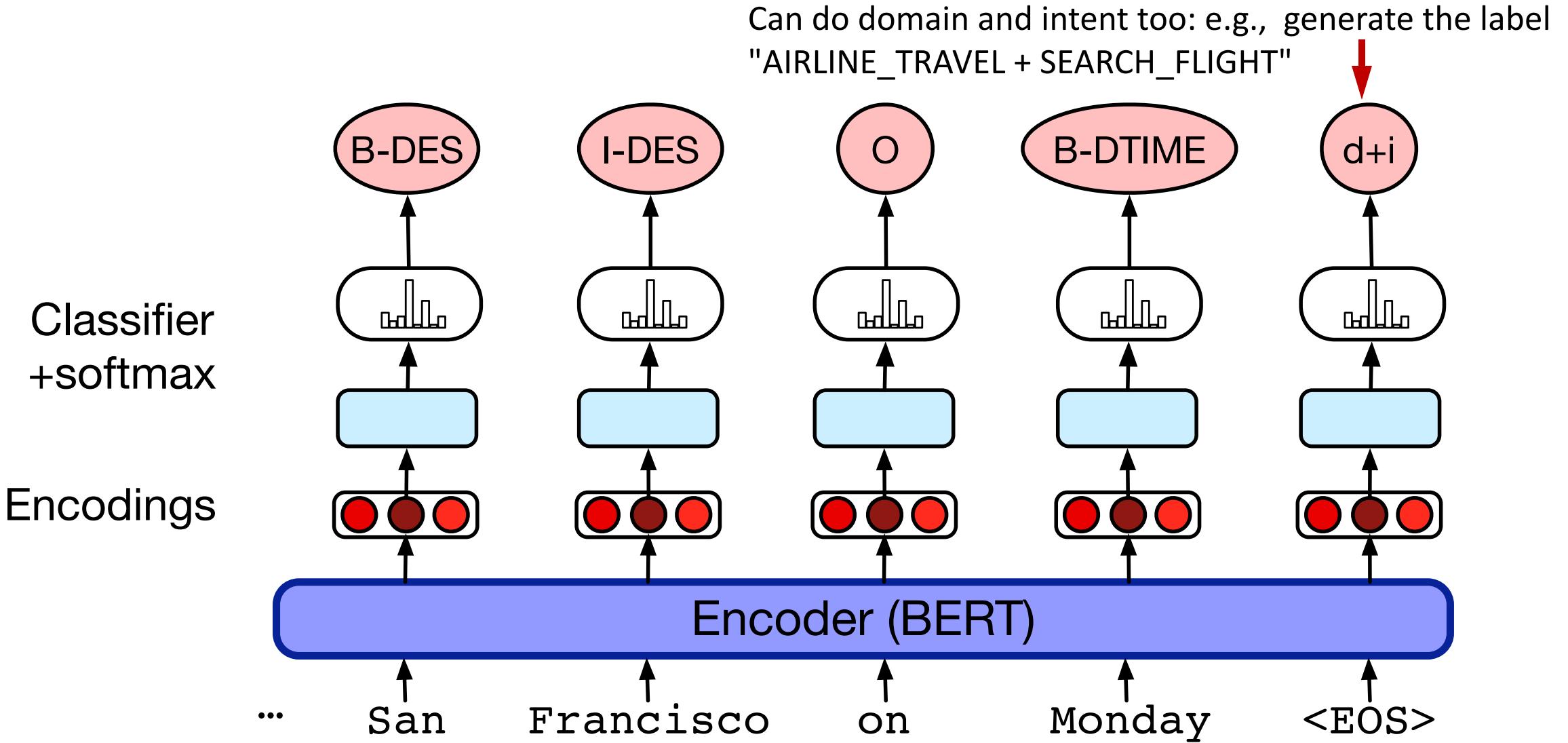
# Slot filling as sequence labeling: BIO tagging

- The **BIO tagging** paradigm
- Idea: Train a classifier to label each input word with a tag that tells us what slot (if any) it fills

0	0	0	0	B-DES	I-DES	0	B-DEPTIME	I-DEPTIME	0	
I	want	to	fly	to	San	Francisco	on	Monday	afternoon	please

- We create a B and I tag for each slot-type
- And convert the training data to this format

# Slot filling using contextual embeddings



# Once we have the BIO tag of the sentence

0	0	0	0	0	B-DES	I-DES	0	B-DEPTIME	I-DEPTIME	0
I	want	to	fly	to	San	Francisco	on	Monday	afternoon	please

- We can extract the filler string for each slot
- And then normalize it to the correct form in the ontology
- Like "SFO" for San Francisco
- Using homonym dictionaries (SF=SFO=San Francisco)

# The task of dialogue state tracking

User: I'm looking for a cheaper restaurant  
`inform(price=cheap)`

System: Sure. What kind - and where?

User: Thai food, somewhere downtown  
`inform(price=cheap, food=Thai, area=centre)`

System: The House serves cheap Thai food

User: Where is it?  
`inform(price=cheap, food=Thai, area=centre); request(address)`

System: The House is at 106 Regent Street

# Evaluating Dialog State Tracking

- Belief State or Dialog State in a Task-oriented Dialog
  - Collection of the slot-value pairs mentioned so far
  - E.g. (location:downtown, price: cheap, food-type: Indian, table-for: 3)
  - Cumulative in nature
- Evaluation:
  - Compare Ground Truth ( $G_t$ ) and Predicted Dialog States ( $P_t$ ), at the end of turn t
  - **Joint Goal Accuracy (JGA):** Checks whether  $(G_t == P_t)$ 
    - Boolean, and strict
  - **Slot Accuracy (SA):** Number of slots whose values match correctly in  $G_t$  and  $P_t$ 
    - individually compares each (domain, slot, value)
  - **Flexible Goal Accuracy (FGA):** Penalize for errors made in this turn. Discount the penalty for errors made in earlier turns.

# Dialogue Policy

- At turn  $i$  predict action  $A_i$  to take, given entire history:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in A} P(A_i | (A_1, U_1, \dots, A_{i-1}, U_{i-1}))$$

- Simplify by just conditioning on the current dialogue state (filled frame slots) and the last turn and turn by system and user:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in A} P(A_i | \text{Frame}_{i-1}, A_{i-1}, U_{i-1})$$

# Policy example: Confirmation and Rejection

- Dialogue systems make errors
- So they to make sure they have understood user
- Two important mechanisms:
- **confirming** understandings with the user
- **rejecting** utterances that the system is likely to have misunderstood.
- Some other policies:
  - EXECUTE: Send query to DB and answers user based on the response from DB
  - CONFIRM: Clarifies the intent with the user
  - ELICIT: Asks for missing information

S: Which city do you want to leave from?

U: Baltimore.

S: Do you want to leave from Baltimore?

U: Yes.

U: I'd like to fly from Denver Colorado to New York City on September twenty first in the morning on United Airlines

S: Let's see then. I have you going from Denver Colorado to New York on September twenty first. Is that correct?

U: Yes

Explicit Confirmation

U: I want to travel to Berlin

S: When do you want to travel to Berlin?

U2: Hi I'd like to fly to Seattle Tuesday Morning

A3: Traveling to Seattle on Tuesday, August eleventh in the morning.  
Your full name?

Implicit Confirmation

# Confirmation strategy tradeoffs

- Explicit confirmation makes it easier for users to correct the system's misrecognitions since a user can just answer "no" to the confirmation question.
- But explicit confirmation is also awkward and increases the length of the conversation (Danieli and Gerbino 1995, Walker et al. 1998).

# Progressive prompting for rejection

Don't just repeat the question "When would you like to leave?"  
Give user guidance about what they can say:

System: When would you like to leave?

Caller: Well, um, I need to be in New York in time for the first World Series game.

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

Caller: I wanna go on October fifteenth.

# Using confidence to decide whether to confirm:

- ASR or NLU systems can assign a **confidence** value, indicating how likely they are that they understood the user.
  - Acoustic log-likelihood of the utterance
  - Prosodic features
  - Ratio of score of best to second-best interpretation
- Systems could use set confidence thresholds:

$< \alpha$	low confidence	reject
$\geq \alpha$	above the threshold	confirm explicitly
$\geq \beta$	high confidence	confirm implicitly
$\geq \gamma$	very high confidence	don't confirm at all

# Natural Language Generation

- NLG in information-state architecture modeled in two stages:
  - **content planning** (what to say)
  - **sentence realization** (how to say it).
- We'll focus on sentence realization here.

# Sentence Realization

- Assume content planning has been done by the dialogue policy
  - Chosen the dialogue act to generate
  - Chosen some attributes (slots and values) that the planner wants to say to the user
    - Either to give the user the answer, or as part of a confirmation strategy)

# Soft Input and Output for Sentence Realizer

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

- 1 Au Midi is in Midtown and serves French food.
- 2 There is a French restaurant in Midtown called Au Midi.

```
recommend(restaurant name= Loch Fyne, neighborhood = city  
centre, cuisine = seafood)
```

- 3 Loch Fyne is in the City Center and serves seafood food.
  - 4 There is a seafood restaurant in the City Centre called Loch Fyne.
-

# Sentence Realization

- Training data is hard to come by
  - Don't see each restaurant in each situation
- Common way to improve generalization:
  - **Delexicalization:** replacing words in the training set that represent slot values with a generic placeholder token:

```
recommend(restaurant name= Au Midi, neighborhood = midtown,  
cuisine = french
```

- 1 Au Midi is in Midtown and serves French food.
- 2 There is a French restaurant in Midtown called Au Midi.

# Sentence Realization

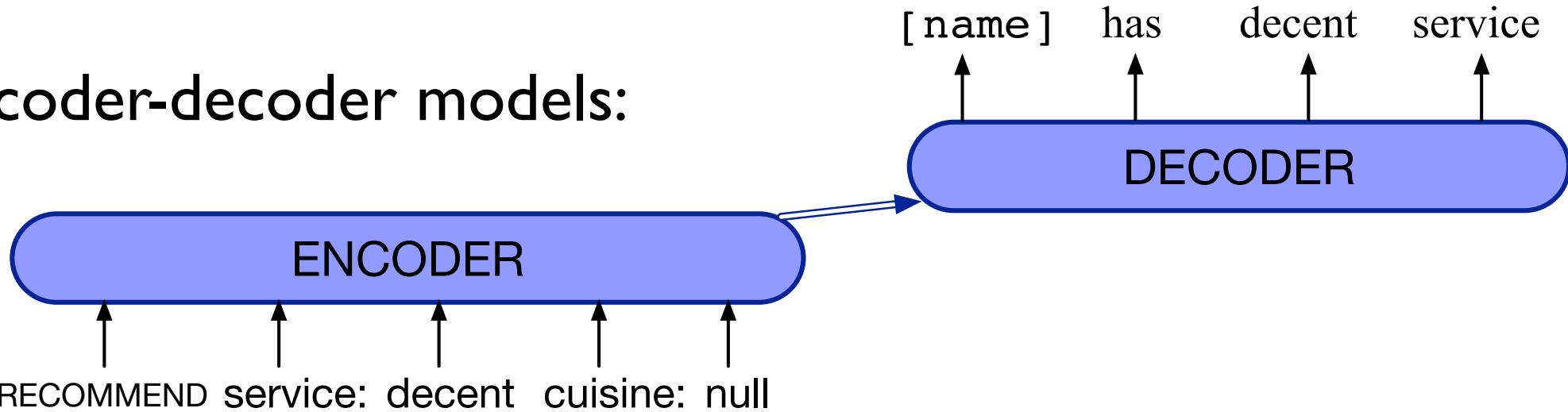
- Training data is hard to come by
  - Don't see each restaurant in each situation
- Common way to improve generalization:
  - **Delexicalization:** replacing words in the training set that represent slot values with a generic placeholder token:

```
recommend(restaurant name= Au Midi, neighborhood = midtown,  
cuisine = french
```

- 1 `restaurant_name` is in `neighborhood` and serves `cuisine` food.
- 2 There is a `cuisine` restaurant in `neighborhood` called `restaurant_name`.

# Sentence Realization: mapping from frames to delexicalized sentences

- Encoder-decoder models:



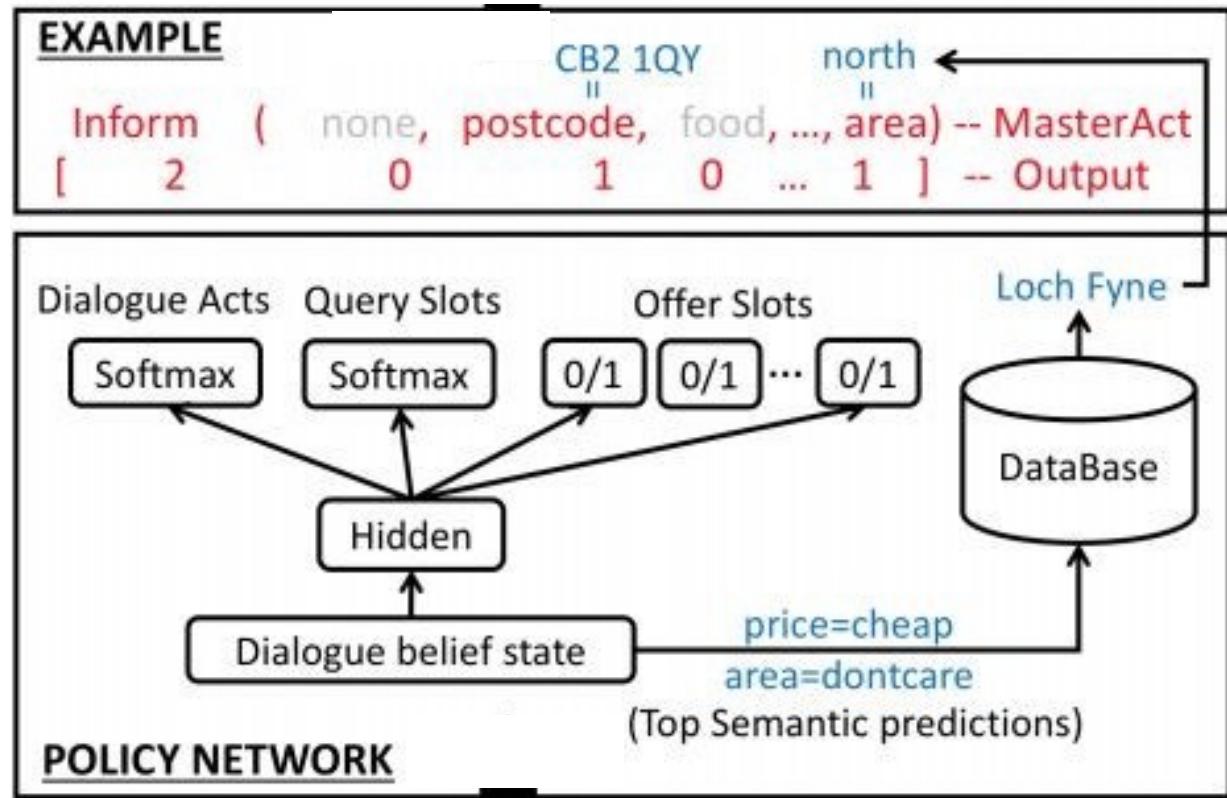
- Output:

restaurant\_name has decent service

- Relexicalize to:

- Au Midi has decent service

# Neural Dialog Management



Dialogue belief state: encodes the understood user intents + dialogue history

Policy Network: 1 hidden layer ( $\tanh$ ), output layer with 2 softmax partitions, 3 sigmoid partitions

Dialogue Acts: {request, offer, inform, select, bye}

Query slots: {food, price-range, area, none}

Offer slots: {Area, phone, postcode}

# Repetition in Neural chatbots

Baseline mutual information model (Li et al. 2015)	Proposed reinforcement learning model
A: Where are you going? (1)	A: Where are you going? (1)
B: I'm going to the restroom. (2)	B: I'm going to the police station. (2)
A: See you later. (3)	A: I'll come with you. (3)
B: See you later. (4)	B: No, no, no, no, you're not going anywhere. (4)
A: See you later. (5)	A: Why? (5)
B: See you later. (6)	B: I need you to stay here. (6)
...	A: I don't know what you are talking about. (7)
...	...
A: how old are you? (1)	A: How old are you? (1)
B: I'm 16. (2)	B: I'm 16. Why are you asking? (2)
A: 16? (3)	A: I thought you were 12. (3)
B: I don't know what you are talking about. (4)	B: What made you think so? (4)
A: You don't know what you are saying. (5)	A: I don't know what you are talking about. (5)
B: I don't know what you are talking about. (6)	B: You don't know what you are saying. (6)
A: You don't know what you are saying. (7)	...
...	...

Table 1: **Left Column:** Dialogue simulation between two agents using a 4-layer LSTM encoder-decoder trained on the OpenSubtitles dataset. The first turn (index 1) is input by the authors. Then the two agents take turns conversing, taking as input the other agent's prior generated turn. The output is generated using the mutual information model (Li et al., 2015) in which an N-best list is first obtained using beam search based on  $p(t|s)$  and reranked by linearly combining the backward probability  $p(s|t)$ , where  $t$  and  $s$  respectively denote targets and sources. **Right Column:** Dialogue simulated using the proposed reinforcement learning model. The new model has more forward-looking utterances (questions like “Why are you asking?” and offers like “I’ll come with you”) and lasts longer before it falls into conversational black holes.

# Deep Reinforcement Learning (DRL) for Dialogue Generation

- **State:** concatenation of the previous two dialogue turns  
 $[p_i, q_i]$  → Input to the encoder
- **Action:** dialogue utterance to generate (infinite action space)
- **Policy:**  $p_{RL}(p_{i+1}|p_i, q_i)$ ; stochastic; parameters of the encoder-decoder
- **Reward:** Easy to answer, non-repetitive, semantic coherence

25

# Reward #1: Ease of Answering

Ease of answering = - (likelihood of dull response)

$\mathbb{S} = \{\text{"I don't know. I'm not sure"}, \dots\}$

$$r_1 = -\frac{1}{N_{\mathbb{S}}} \sum_{s \in \mathbb{S}} \frac{1}{N_s} \log p_{\text{seq2seq}}(s|a)$$

$N_{\mathbb{S}}$  = Cardinality of  $\mathbb{S}$

$N_s$  = length of dull response  $s$

$p_{\text{seq2seq}}$  = likelihood given by the SEQ2SEQ model

# Reward #2: Information Flow

- High information flow = avoid repetitive/similar responses

$$r_2 = -\log \cos(h_{p_i}, h_{p_{i+1}}) = -\log \cos \frac{h_{p_i} \cdot h_{p_{i+1}}}{\|h_{p_i}\| \|h_{p_{i+1}}\|}$$

$h_p$  = encoder representation of utterance p

# Reward #3: Semantic Coherence

- High semantic coherence = high mutual information between two consecutive answers

$$r_3 = \frac{1}{N_a} \log p_{\text{seq2seq}}(a|q_i, p_i) + \frac{1}{N_{q_i}} \log p_{\text{seq2seq}}^{\text{backward}}(q_i|a)$$

# Total Reward

$$r(a, [p_i, q_i]) = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3$$

where  $\lambda_1 + \lambda_2 + \lambda_3 = 1$

Overall strategy:

- Pre-train SEQ2SEQ with MLE objective
- Let two virtual agents talk to each other and optimize the policy by maximizing the expected reward (use policy gradient methods)  $p_{RL}(p_{i+1}|p_i, q_i)$

# DRL in Retrieval-based Dialogue

- Training:
  - Phase 1: Supervised learning on AMT corpora of 720 dialogues, maximize likelihood of data
  - Phase 2: Reinforcement Learning; find policy that maximizes expected reward of a dialogue with T turns

$$J(\theta) = E \left[ \sum_{t=1}^T \gamma^t r(s_t, a_t) \middle| \pi_\theta \right]$$

# DRL in Retrieval-based Dialogue

- Training:
  - Phase 1: Supervised learning on AMT corpora of 720 dialogues, maximize likelihood of data
  - Phase 2: Reinforcement Learning; find policy that maximizes expected reward of a dialogue with T turns


$$J(\theta) = E \left[ \sum_{t=1}^T \gamma^t r(s_t, a_t) \middle| \pi_\theta \right]$$

**Policy Gradient  
Methods**

# Policy Gradient Methods

- A class of RL methods (Lecture 7a)
- Problem: Maximize  $E[ R | \pi_\theta ]$
- Intuitions: collect a bunch of trajectories using  $\pi_\theta$ , and
  - Make the good trajectories more probable
  - Make the good actions more probable

# Generating clarification questions

- User: What do you have going to UNKNOWN WORD on the 5th?
- System: Going where on the 5th?
- The system repeats “going” and “on the 5th” to make it clear which aspect of the user’s turn the system needs to be clarified
- Methods for generating clarification questions:
  - Rules like 'replace “going to UNKNOWN WORD” with “going where”'
  - Classifiers that guess which slots were misrecognized

# Evaluating chatbots and task-based dialogue

- Task-based dialogue:
  - mainly by measuring task performance
- Chatbots or Open-Domain Dialogs:
  - Automated metrics: BLEU, Dist-1/2/3, Perplexity
    - Not very reliable in such one-to-many setup
  - Human evaluation is important

# **Chatbots are evaluated by humans**

- **Participant evaluation:** The human who talked to the chatbot assigns a score
- **Observer evaluation:** third party who reads a transcript of a human/chatbot conversation assigns a score.

# Participant evaluation

- Human chats with model for 6 turns and rates 8 dimensions of quality:
- **avoiding repetition, interestingness, making sense, fluency, listening, inquisitiveness, humanness, engagingness,**
- **(1) Avoiding Repetition:** How repetitive was this user?
  - Repeated themselves over and over
  - Sometimes said the same thing twice
  - Always said something new
- **(3) Making sense:** How often did this user say something which didn't make sense?
  - Never made any sense
  - Most responses didn't make sense
  - Some responses didn't make sense
  - Everything made perfect sense
- **(8) Engagingness:** How much did you enjoy talking to this user?
  - Not at all
  - A little
  - Somewhat
  - A lot

# Observer evaluation: acute-eval

- Annotators look at two conversations (A + B) and decide which is better:
- **Engagingness:** Who would you prefer to talk to for a long conversation?
- **Interestingness:** If you had to say one of these speakers is interesting and one is boring, who would you say is more interesting?
- **Humanness:** Which speaker sounds more human?
- **Knowledgeable:** If you had to say that one speaker is more knowledgeable and one is more ignorant, who is more knowledgeable?

# ACUTE-EVAL

<p>Hi how are you today</p> <p>Hi, I'm pretty good! Just listening to some aerosmith, they're my fave :) what're you up to?</p> <p>I am listening to some italian music</p> <p>Italian music, nice! What do you do for work?</p> <p>I'm a professional wrestler</p> <p>Oh wow! I'm a construction worker, I've been doing it since I was 16.</p> <p>What do you do for a living</p> <p>I'm a construction worker haha, just told you! What's your favorite type of food?</p> <p>My favorite is garlic bread</p> <p>Yum! I love mexican food, so good :) do you have any hobbies?</p> <p>I like to workout and go to the gym</p> <p>We're a bit different- I love watching nascar and ufc. They're so fun!</p>	<p>Hello there, how are you?</p> <p>I am doing great. How are you?</p> <p>I am great, I did something crazy for me and colored my hair blue!</p> <p>I have a daughter and a son who also love blue colored balls. You should meet them</p> <p>Well that's neat, I got a new car my mother gave so maybe I could see them!</p> <p>It is a beautiful city. And, I try to be... Just cannot afford a bigger house atm.</p> <p>I am sorry to hear that, I feel bad going out of town for spring break now.</p> <p>Ok. I'm going to school in the spring for casino manager</p> <p>Well I turn 29 next week, I wonder if that is a good age to apply as one.</p> <p>My grandmother just died from lung cancer, sucks</p>
<p><b>Who would you prefer to talk to for a long conversation?</b></p> <p><input type="radio"/> I would prefer to talk to Speaker 1    <input type="radio"/> I would prefer to talk to Speaker 2</p> <p><b>Please provide a brief justification for your choice (a few words or a sentence)</b></p> <p>Please enter here...</p>	

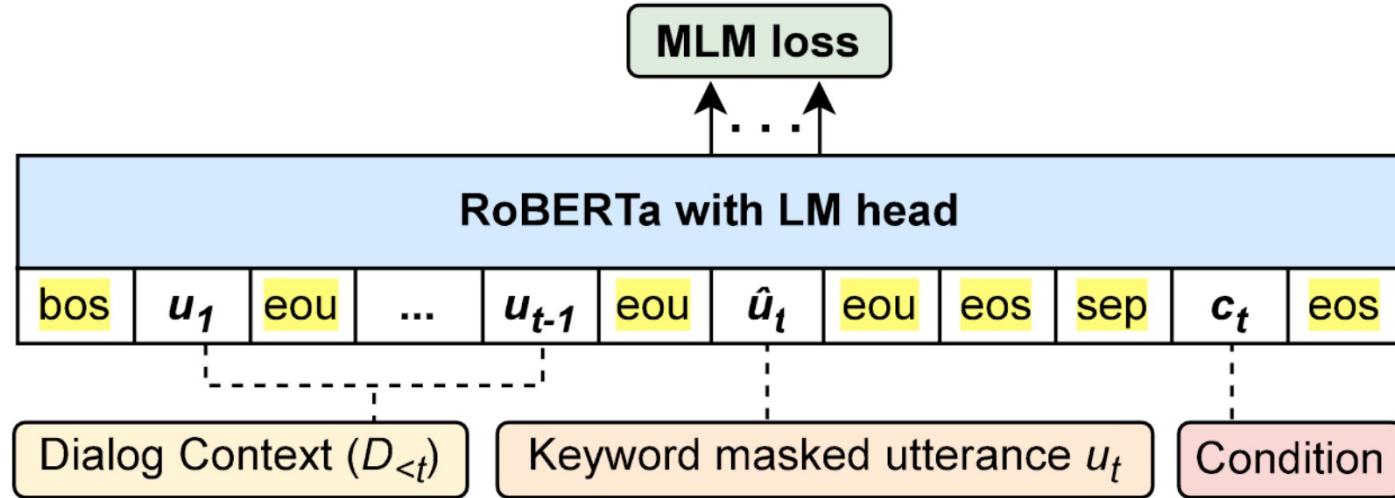
# Automatic evaluation is an open problem

- Automatic evaluation methods (like the BLEU scores used for Machine Translation) are generally not used for chatbots.
- They correlate poorly with human judgements.
- One current research direction: **Adversarial Evaluation**
- Inspired by the Turing Test
- train a ``Turing-like'' classifier to distinguish between human responses and machine responses.
- The more successful a dialogue system is at fooling the evaluator, the better the system.

# More fine-grained metrics: User Satisfaction Survey

<b>TTS Performance</b>	Was the system easy to understand ?
<b>ASR Performance</b>	Did the system understand what you said?
<b>Task Ease</b>	Was it easy to find the message/flight/train you wanted?
<b>Interaction Pace</b>	Was the pace of interaction with the system appropriate?
<b>User Expertise</b>	Did you know what you could say at each point?
<b>System Response</b>	How often was the system sluggish and slow to reply to you?
<b>Expected Behavior</b>	Did the system work the way you expected it to?
<b>Future Use</b>	Do you think you'd use the system in the future?

# Dial-M: Reference-Free evaluation for dialog systems



$$s = \frac{1}{|K_t|} \sum_{j=1}^{|K_t|} \left( \frac{1}{|T_{t,j}|} \sum_{y \in T_{t,j}} -\log p(y | D_{<t}, \hat{u}_{t,j}, c_t) \right)$$

# Other Heuristics

- **Efficiency cost:**
  - total elapsed time for the dialogue in seconds,
  - the number of total turns or of system turns
  - total number of queries
  - “turn correction ratio”: % of turns that were used to correct errors
- **Quality cost:**
  - number of ASR rejection prompts.
  - number of times the user had to barge in

# Safety

- Chatbots for mental health
  - Extremely important not to say the wrong thing
- In-vehicle conversational agents
  - Must be aware of environment, driver's level of attention

# Abuse and Representation Harm: The case of Microsoft Tay

- Experimental Twitter chatbot launched in 2016
  - given the profile personality of an 18- to 24-year-old American woman
  - could share horoscopes, tell jokes,
  - asked people to send selfies
  - used informal language, slang, emojis, and GIFs,
  - Designed to learn from users (IR-based)

# The case of Microsoft Tay

- Immediately Tay turned offensive and abusive
  - Obscene and inflammatory tweets
  - Nazi propaganda, conspiracy theories
  - Began harassing women online
  - Reflecting racism and misogyny of Twitter users
- Microsoft took Tay down after 16 hours
- Lessons:
  - User response must be considered in the design phase

# Bias in training datasets

- Henderson *et al.* ran hate-speech and bias detectors on standard training sets for dialogue systems:
  - Twitter, Reddit, other dialogue datasets
- Found bias and hate-speech
  - In training data
  - In dialogue models trained on the data

# Privacy: Training on user data

- Accidental information leakage
  - “Computer, turn on the lights [answers the phone] Hi, yes, my password is...”
  - Henderson show in simulation that this leakage can occur.
- Intentional information leakage
  - Dialogue systems that are designed to send user data to developer or advertisers
  - Important to consider privacy-preserving dialogue systems

Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. 2018. Ethical Challenges in Data-Driven Dialogue Systems. In 2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES '18),

Campagna, Giovanni, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S. Lam. "Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant." In Proceedings of the 26th International Conference on World Wide Web, pp. 341-350. 2017.