Amirkabir University of Technology
(Tehran Polytechnic)

Natural Language Processing

Lecture 21: Dialogue Systems

Amirkabir University of Technology
Dr Momtazi

Outline

- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
  - Rule-base Models
  - Corpus-based
    - Retrieval Methods
    - Generation Methods
- Task-based Chatbots
  - GUS
  - DSA
- Evaluation

Dialogue Systems

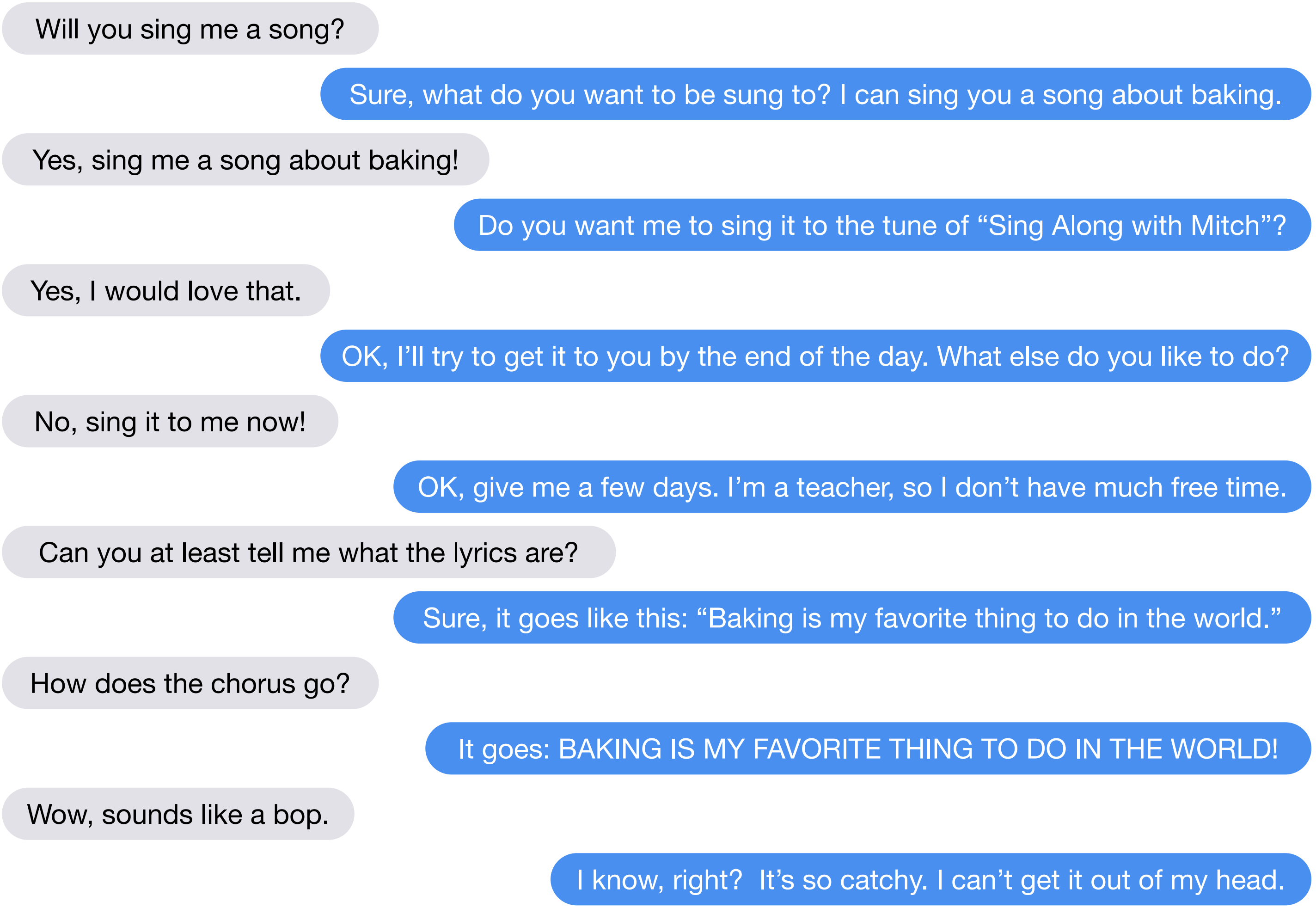
- Known as
  - dialogue systems
  - conversational agents
- Personal Assistants on phones or other devices
  - SIRI, Alexa, Cortana, Google Assistant
- Playing music, setting timers and clocks
- Chatting for fun
- Booking travel reservations
- Clinical uses for mental health

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

Facebook’s 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?

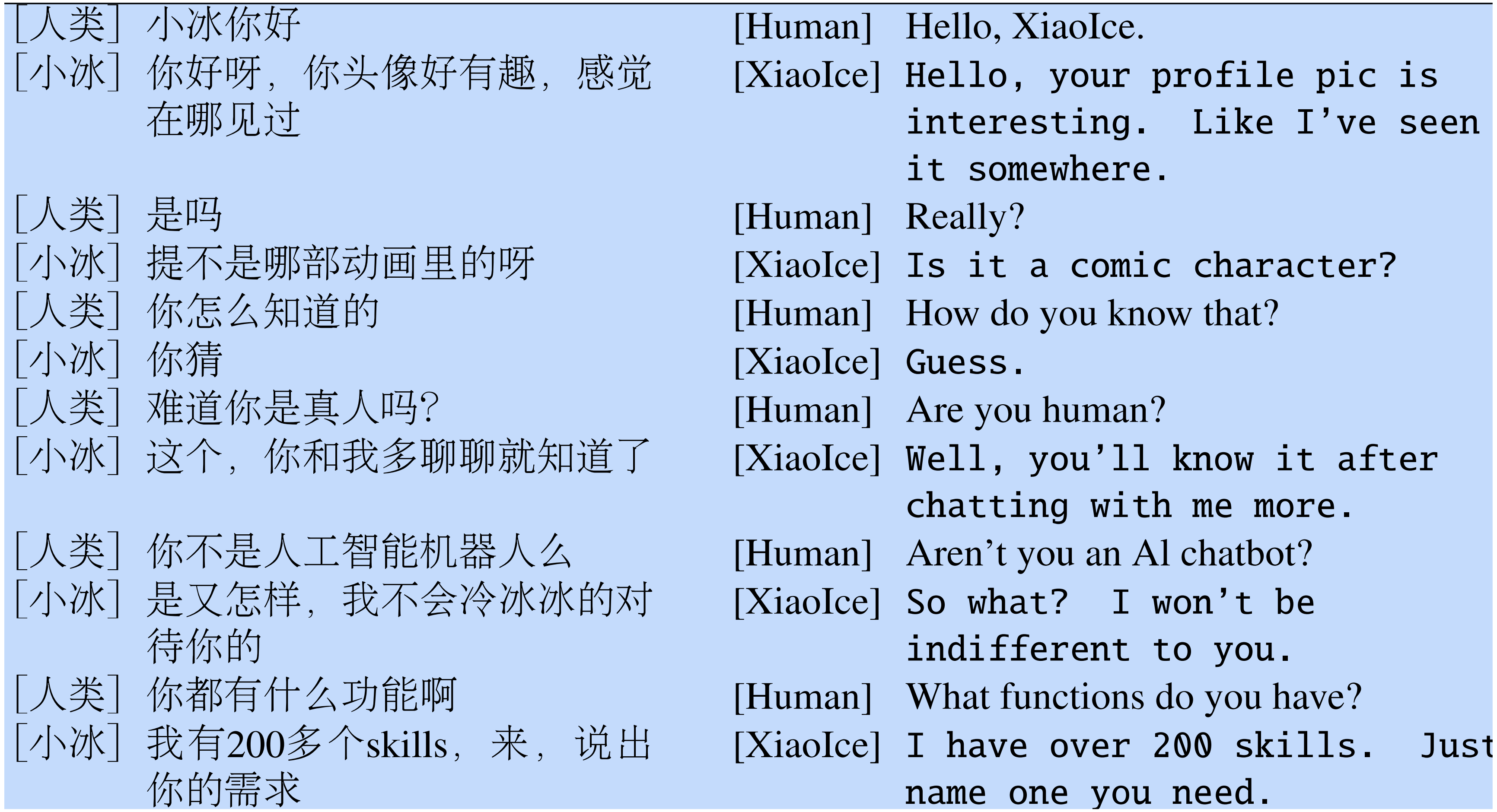
It goes: BAKING IS MY FAVORITE THING TO DO IN THE WORLD!

Wow, sounds like a bop.

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

BlenderBot (Roller et al. 2020)

Microsoft’s XiaoIce System



[人类]	小冰你好	[Human]	Hello, XiaoIce.
[小冰]	你好呀，你头像好有趣，感觉在哪见过	[XiaoIce]	Hello, your profile pic is interesting. Like I’ve seen it somewhere.
[人类]	是吗	[Human]	Really?
[小冰]	提不是哪部动画里的呀	[XiaoIce]	Is it a comic character?
[人类]	你怎么知道的	[Human]	How do you know that?
[小冰]	你猜	[XiaoIce]	Guess.
[人类]	难道你是真人吗?	[Human]	Are you human?
[小冰]	这个，你和我多聊聊就知道了	[XiaoIce]	Well, you’ll know it after chatting with me more.
[人类]	你不是人工智能机器人么	[Human]	Aren’t you an AI chatbot?
[小冰]	是又怎样，我不会冷冰冰的对待你的	[XiaoIce]	So what? I won’t be indifferent to you.
[人类]	你都有什么功能啊	[Human]	What functions do you have?
[小冰]	我有200多个skills，来，说出你的需求	[XiaoIce]	I have over 200 skills. Just name one you need.

XiaoIce (Zhou et al., 2020)

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

The Frame

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

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

Chatbot Architectures

- **Rule-based**
    - Pattern-action rules (ELIZA)
    - + A mental model (PARRY):

      The first system to pass the Turing Test!

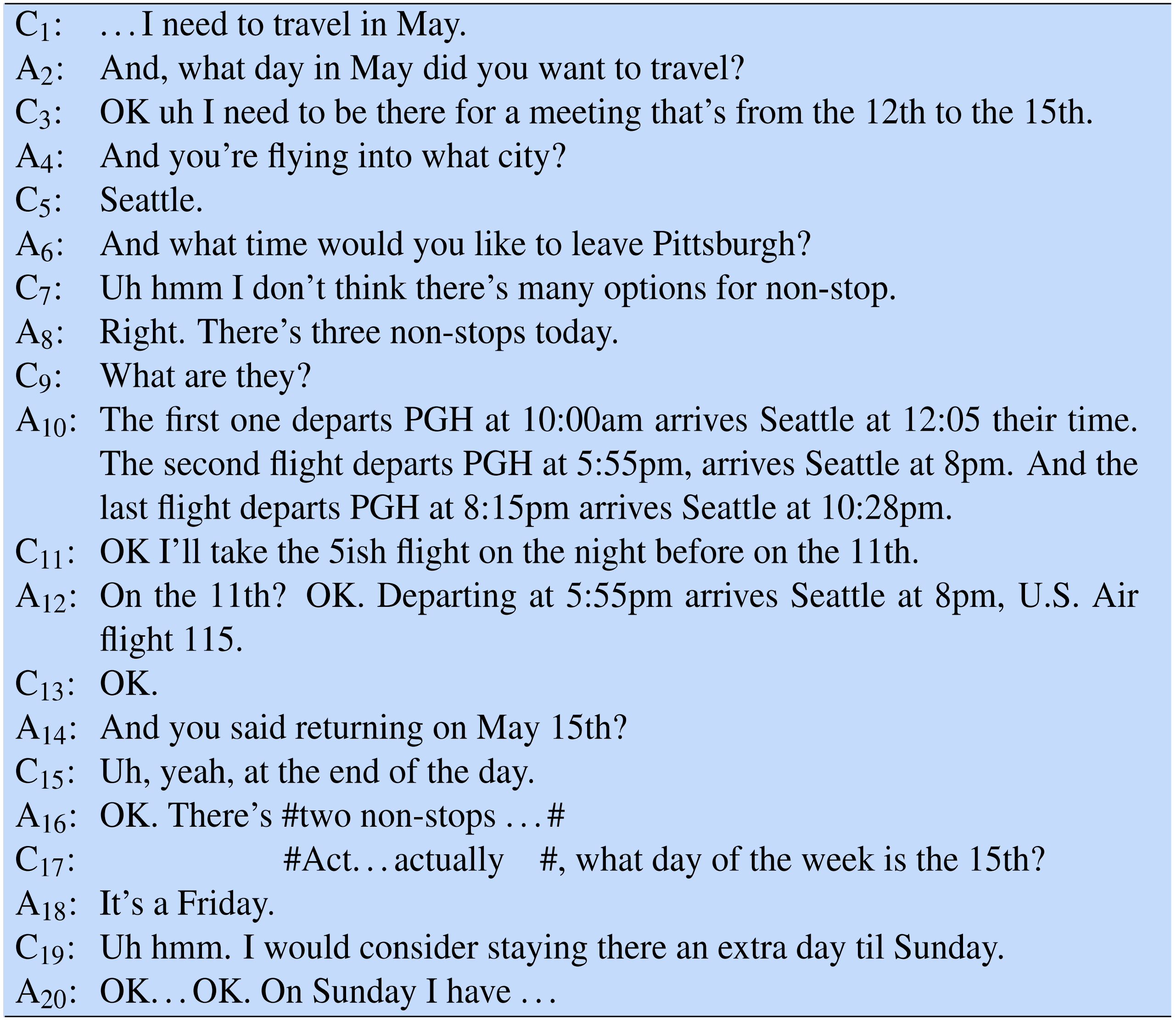
- **Corpus-based**
    - Information Retrieval (XiaoIce)
    - Neural encoder-decoder (BlenderBot)

Outline

- Introduction
- **Properties of Human Conversation**
- Conversational Chatbots
- Task-based Chatbots
- Evaluation

Properties of Human Conversation

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



C_1: . . . I need to travel in May.

A_2: And, what day in May did you want to travel?

C_3: OK uh I need to be there for a meeting that's from the 12th to the 15th.

A_4: And you're flying into what city?

C_5: Seattle.

A_6: And what time would you like to leave Pittsburgh?

C_7: Uh hmm I don't think there's many options for non-stop.

A_8: Right. There's three non-stops today.

C_9: What are they?

A_{10}: 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_{11}: OK I'll take the 5ish flight on the night before on the 11th.

A_{12}: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.

C_{13}: OK.

A_{14}: And you said returning on May 15th?

C_{15}: Uh, yeah, at the end of the day.

A_{16}: OK. There's #two non-stops . . . #

C_{17}: #Act. . . actually #, what day of the week is the 15th?

A_{18}: It's a Friday.

C_{19}: Uh hmm. I would consider staying there an extra day til Sunday.

A_{20}: OK. . . OK. On Sunday I have . . .

Main Elements in Conversations

Turn

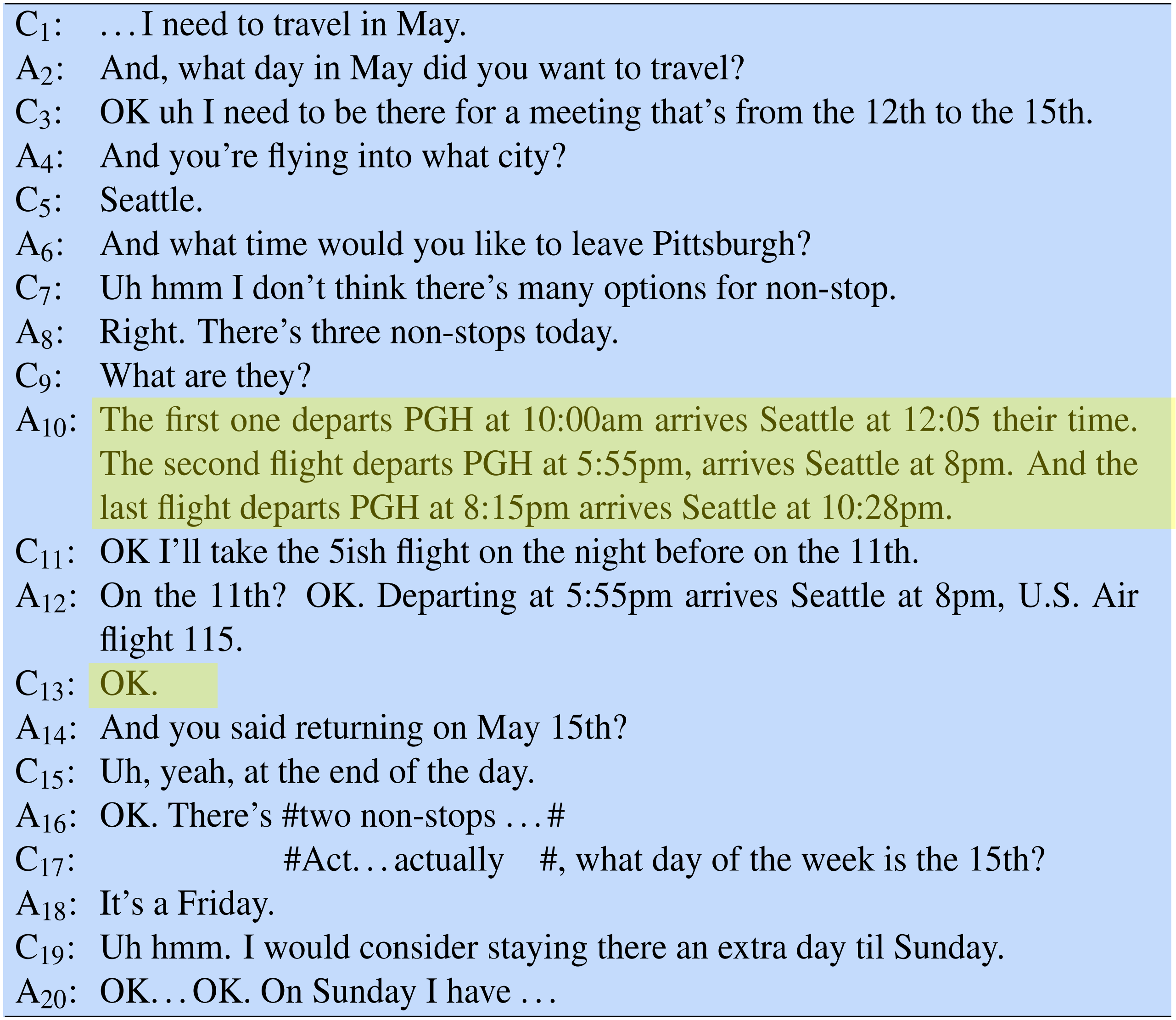
Properties of Human Conversation

Turns

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

Turns

- A turn can consist of
  - a sentence
    - (C1)
  - a short text as a single word
    - (C13)
  - a long text as multiple sentences
    - (A10)



C_1: . . . I need to travel in May.
A_2: And, what day in May did you want to travel?
C_3: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_4: And you're flying into what city?
C_5: Seattle.
A_6: And what time would you like to leave Pittsburgh?
C_7: Uh hmm I don't think there's many options for non-stop.
A_8: Right. There's three non-stops today.
C_9: What are they?
A_{10}: 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_{11}: OK I'll take the 5ish flight on the night before on the 11th.
A_{12}: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_{13}: OK.
A_{14}: And you said returning on May 15th?
C_{15}: Uh, yeah, at the end of the day.
A_{16}: OK. There's #two non-stops . . . #
C_{17}: #Act. . . actually #, what day of the week is the 15th?
A_{18}: It's a Friday.
C_{19}: Uh hmm. I would consider staying there an extra day til Sunday.
A_{20}: OK. . . OK. On Sunday I have . . .

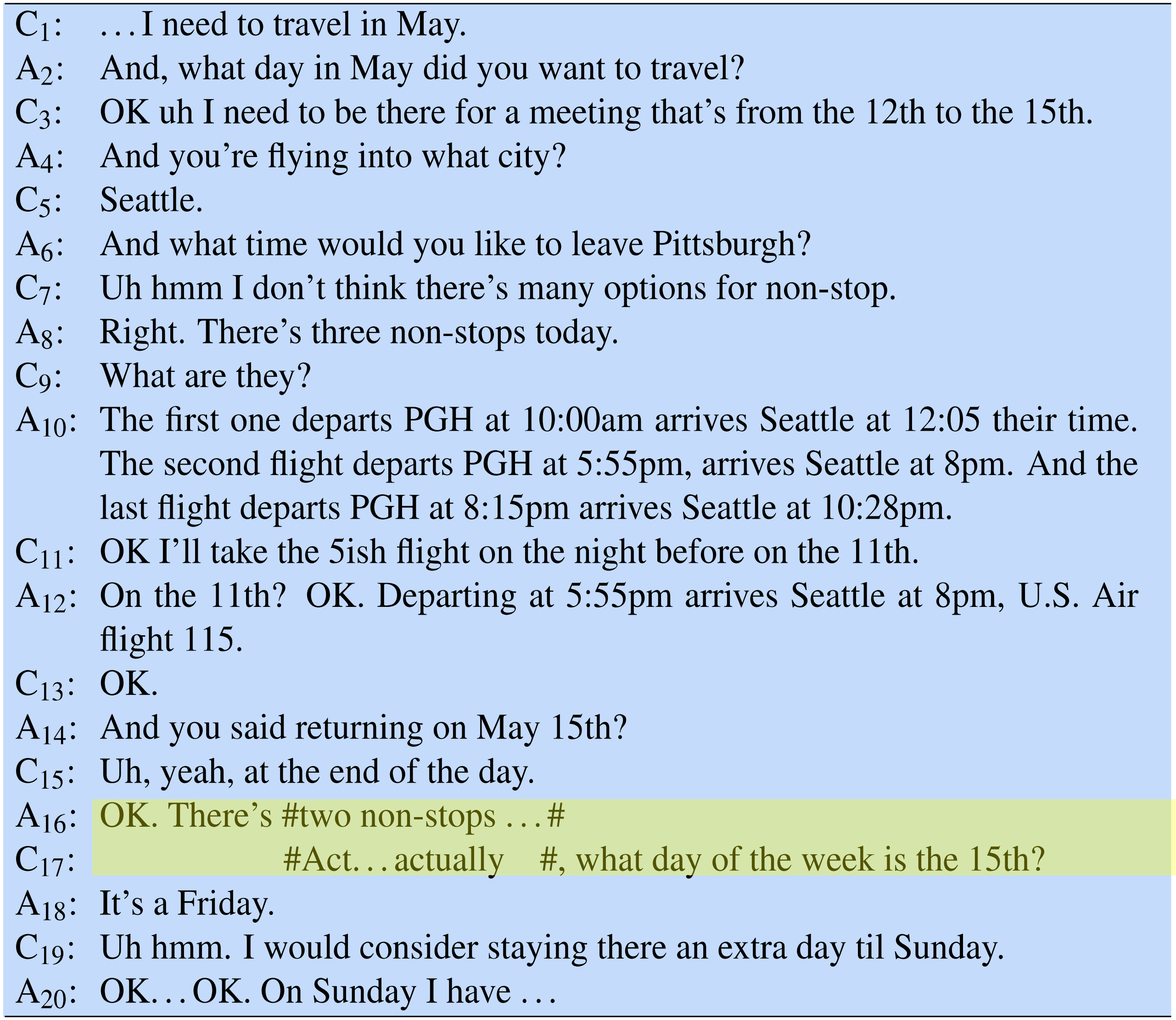
Properties of Human Conversation

Turn-taking issues

- When to take the floor?
- When to yield the floor?

Interruptions

Endpoint Detection



C_1: …I need to travel in May.
A_2: And, what day in May did you want to travel?
C_3: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A_4: And you're flying into what city?
C_5: Seattle.
A_6: And what time would you like to leave Pittsburgh?
C_7: Uh hmm I don't think there's many options for non-stop.
A_8: Right. There's three non-stops today.
C_9: What are they?
A_{10}: 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_{11}: OK I'll take the 5ish flight on the night before on the 11th.
A_{12}: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.
C_{13}: OK.
A_{14}: And you said returning on May 15th?
C_{15}: Uh, yeah, at the end of the day.
A_{16}: OK. There's #two non-stops …#
C_{17}: #Act…actually #, what day of the week is the 15th?
A_{18}: It's a Friday.
C_{19}: Uh hmm. I would consider staying there an extra day til Sunday.
A_{20}: OK…OK. On Sunday I have …

Endpoint Detection

- A system has to know when to stop talking; the client interrupts (in A16 and C17), so the system must know to stop talking (and that the user might be making a correction).
- A system also has to know when to start talking.
    - For example, most of the time in conversation, speakers start their turns almost immediately after the other speaker finishes, without a long pause, because people are able to (most of the time) detect when the other person is about to finish talking.
    - Spoken dialogue systems must also detect whether a user is done speaking, so they can process the utterance and respond.
- This task is quite challenging because of noise and because people often pause in the middle of turns.

Implications for Conversational Agents

Barge-in

- Allowing the user to interrupt

End-pointing

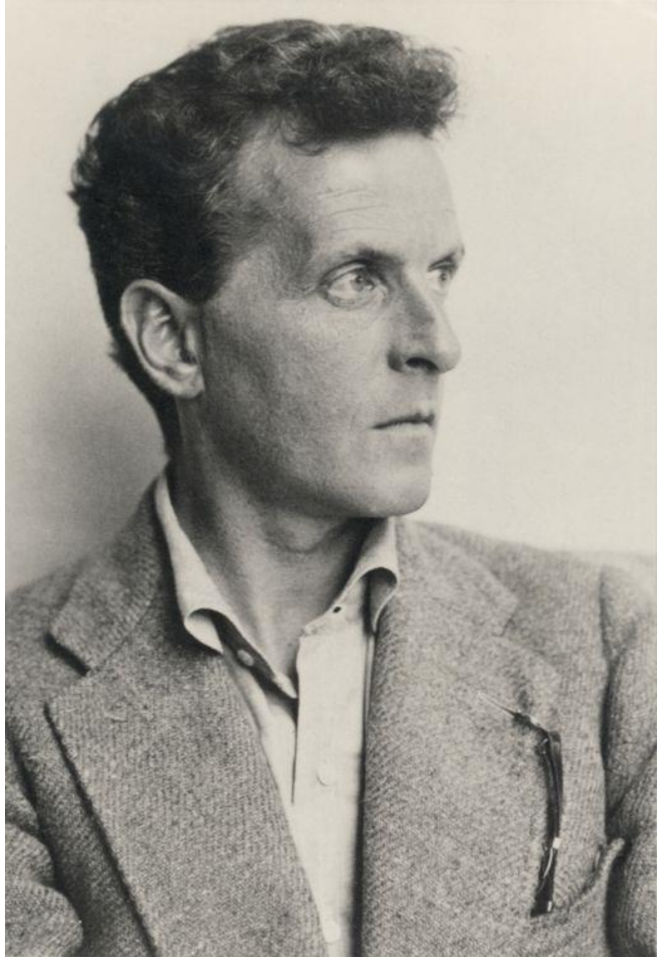
- The task for a speech system of deciding whether the user has stopped talking
- Very hard, since people often pause in the middle of turns

Main Elements in Conversations

Turn

Speech Act

Language as Action



Each turn in a dialogue is a kind of action

Wittgenstein (1953) and Austin (1962)

Speech Acts (aka Dialogue Acts)

- **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

Main Elements in Conversations

Turn

Speech Act

Grounding

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

- Grounding is relevant for human-machine interaction
  - Why do elevator buttons light up?

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!

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!

Cohen et al (2004)

Main Elements in Conversations

Turn

Speech Act

Grounding

Subdialogues

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.

Main Elements in Conversations

- Turn
- Speech Act
- Grounding
- Subdialogues
- Initiative

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.

Conversational Initiative

- 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)

Main Elements in Conversations

- Turn
- Speech Act
- Grounding
- Subdialogues
- Initiative
- Inference and Implicature

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.

Inference

- The speaker seems to expect the hearer to draw certain inferences;
- The speaker is communicating more information than seems to be present in the uttered words.
- Implicature means a particular class of licensed inferences.

- An inference scenario
  - The client mentions a meeting on the 12th,
  - The agent reasons ‘There must be some relevance for mentioning this meeting. What could it be?’.
  - The agent knows that one precondition for having a meeting (at least before Web conferencing) is being at the place where the meeting is held, and therefore that maybe the meeting is a reason for the travel, and if so, then since people like to arrive the day before a meeting, the agent should infer that the flight should be on the 11th.

Outline

- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
  - Rule-base Models
  - **Corpus-based**
    - Retrieval Methods
    - Generation Methods
- Task-based Chatbots
- Evaluation

Two Architectures for Corpus-based Chatbots

- 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

What conversations to draw on?

- Pre-training on larger corpora

- 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)

Outline

- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
  - Rule-base Models
  - **Corpus-based**
    - **Retrieval Methods**
    - Generation Methods
- Task-based Chatbots
- Evaluation

Response by Retrieval

1. Given a user turn *q*, and a training corpus *C* of conversation
2. Find in C the turn *r* that is most similar (text similarity) to *q*

- Corpus-based chatbot algorithms thus draw on algorithms for question answering systems, which similarly focus on single responses while ignoring longer-term conversational goals.

Text Similarity

- Vector Space Model
- Neural Text Matching

Vector Space Model for Retrieval

- Providing a relevance ranking of the documents with respect to a query
  - Assign a score to each query-document pair, say in [0, 1].
  - This score measures how well document and query “match”.
  - Sort documents according to scores

Vector Space Model

- Represent a doc/query by a term vector
  - Term: basic concept, e.g., word
  - Each term defines one dimension
  - N terms define an N-dimensional space
  - Query vector: $q=(x_1, \ldots, x_N)$, x_i is query term weight
  - Doc vector: $d=(y_1, \ldots, y_N)$, y_j is doc term weight

relevance(q,d) ≈ similarity(q,d) = f(q,d)

Vector Space Model with Word Embedding

1. Create a vector for each document
    - TF-IDF
    - Average of Word2Vec
    - Doc2Vec
    - BERT

2. Calculate the cosine similarity between the embedding vectors

Text Similarity

- Vector Space Model
- Neural Text Matching

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 (neural text similarity) to q

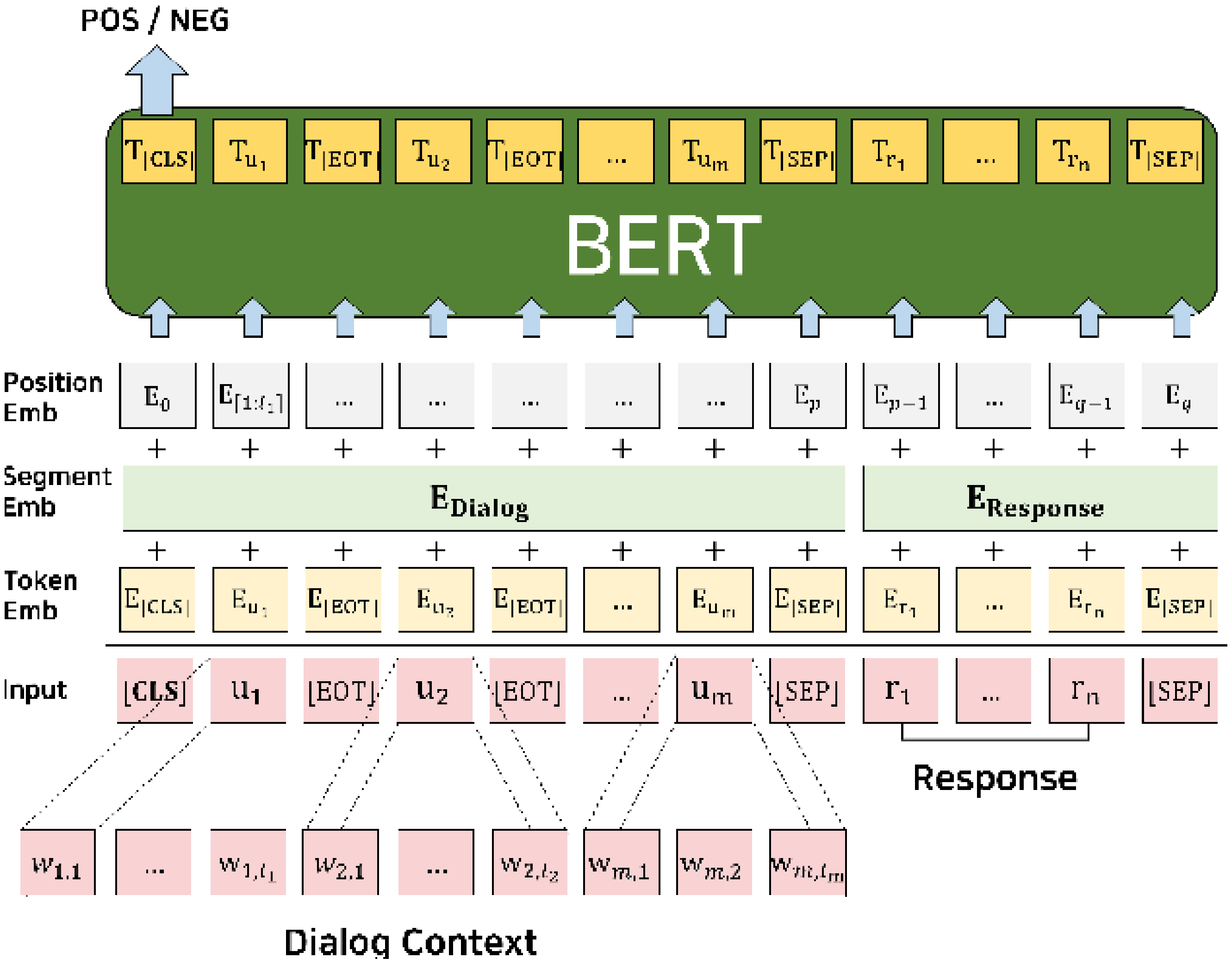
Neural Text Matching

- Learning to match models
- Considering text matching as a supervised objective where pairs of relevant or paraphrased texts are given

- Approaches
    - Representation-based models
    - Interaction-based models
    - Hybrid models

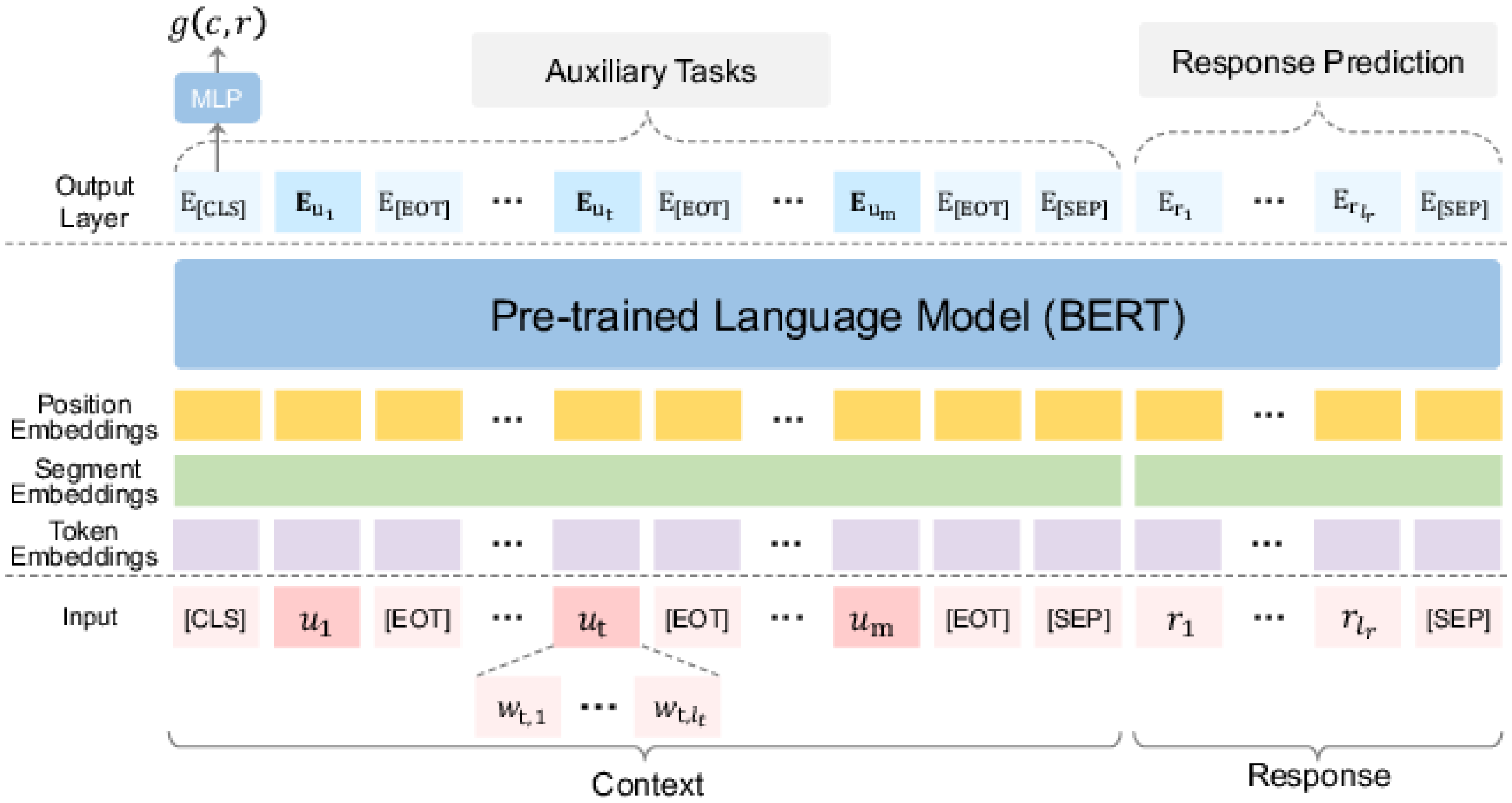
Retrieval Methods with BERT

An Effective Domain Adaptive Post-Training Method for BERT in Response Selection


POS / NEG
$T_{[CLS]}$ T_{u_1} $T_{[EOT]}$ T_{u_2} $T_{[EOT]}$ ... T_{u_m} $T_{[SEP]}$ T_{r_1} ... T_{r_n} $T_{[SEP]}$
BERT
Position Emb
E_0 $E_{[1:l_1]}$ ... ... E_p E_{p-1} ... E_{q-1} E_q
Segment Emb
E_{Dialog}
$E_{Response}$
Token Emb
$E_{[CLS]}$ E_{u_1} $E_{[EOT]}$ E_{u_2} $E_{[EOT]}$ ... E_{u_m} $E_{[SEP]}$ E_{r_1} ... E_{r_n} $E_{[SEP]}$
Input
[CLS] u_1 [EOT] u_2 [EOT] ... u_m [SEP] r_1 ... r_n [SEP]
Response
$w_{1,1}$ ... w_{1,l_1} $w_{2,1}$ ... w_{2,l_2} $w_{m,1}$ $w_{m,2}$ w_{m,l_m}
Dialog Context

Retrieval Methods with BERT

Learning an Effective Context-Response Matching Model with Self-Supervised Tasks for Retrieval-based Dialogues



Outline

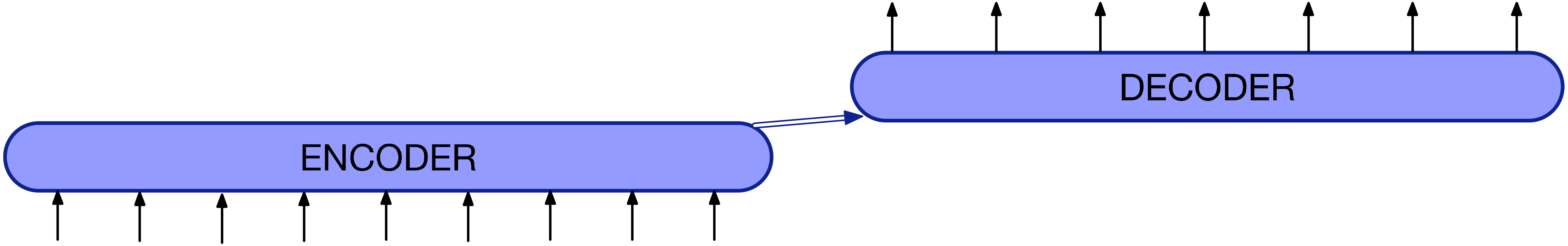
- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
  - Rule-base Models
  - **Corpus-based**
    - Retrieval Methods
    - **Generation Methods**
- Task-based Chatbots
- Evaluation

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 $r_1 ... r_{t-1}$

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

Response by Generation



[U:] I finally got promoted today at work!

[S:] Congrats! That’s great!

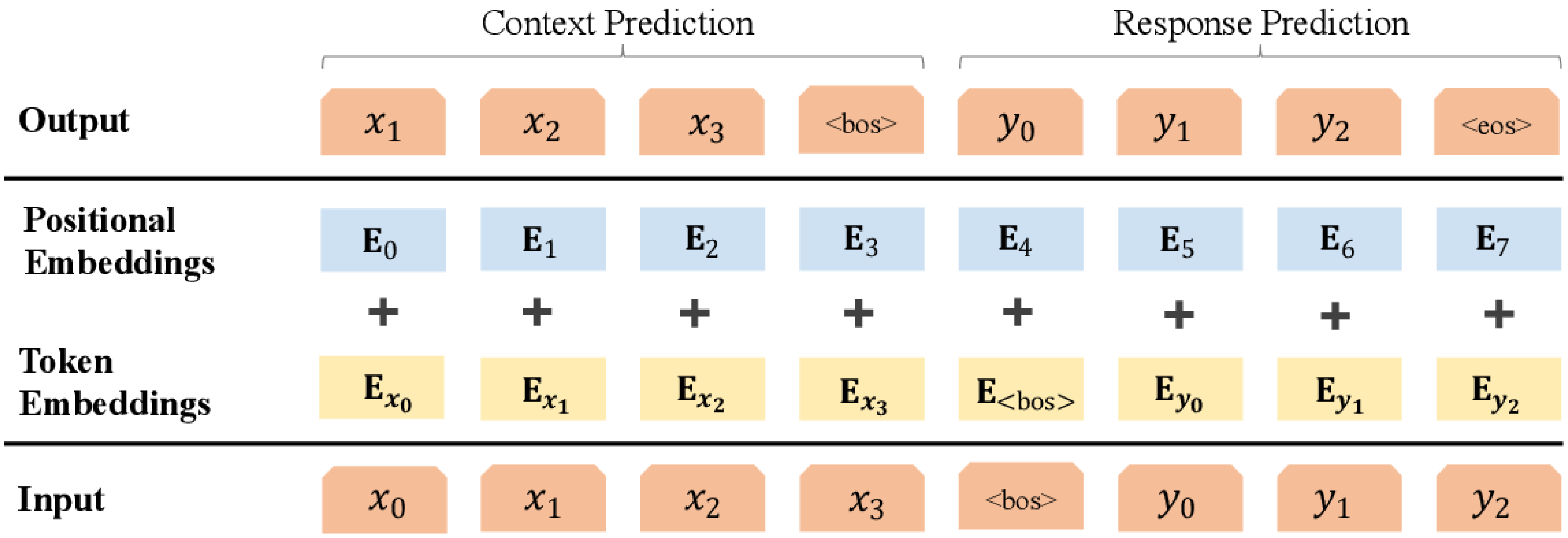
[U:] Thank you! I’ve been trying to get it for a while now!

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)

Generation Methods with GPT

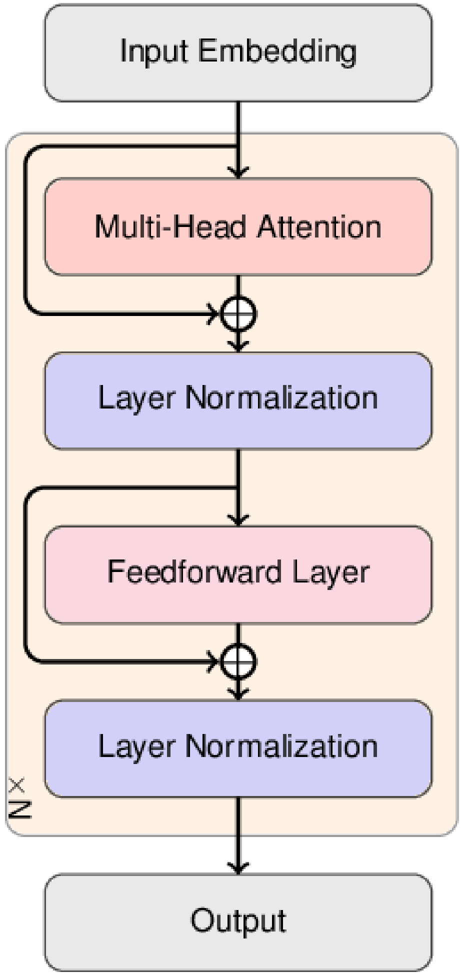
An Empirical Investigation of Pre-Trained Transformer Language Models for Open-Domain Dialogue Generation



	Context Prediction				Response Prediction			
Output	x_1	x_2	x_3	<bos>	y_0	y_1	y_2	<eos>
Positional Embeddings	$\mathbf{E}_0$	$\mathbf{E}_1$	$\mathbf{E}_2$	$\mathbf{E}_3$	$\mathbf{E}_4$	$\mathbf{E}_5$	$\mathbf{E}_6$	$\mathbf{E}_7$
	+	+	+	+	+	+	+	+
Token Embeddings	$\mathbf{E}_{x_0}$	$\mathbf{E}_{x_1}$	$\mathbf{E}_{x_2}$	$\mathbf{E}_{x_3}$	$\mathbf{E}_{\text{<bos>}}$	$\mathbf{E}_{y_0}$	$\mathbf{E}_{y_1}$	$\mathbf{E}_{y_2}$
Input	x_0	x_1	x_2	x_3	<bos>	y_0	y_1	y_2

Generation Methods with GPT

Large-Scale Transfer Learning for Natural Language Generation



Input Embedding

Multi-Head Attention

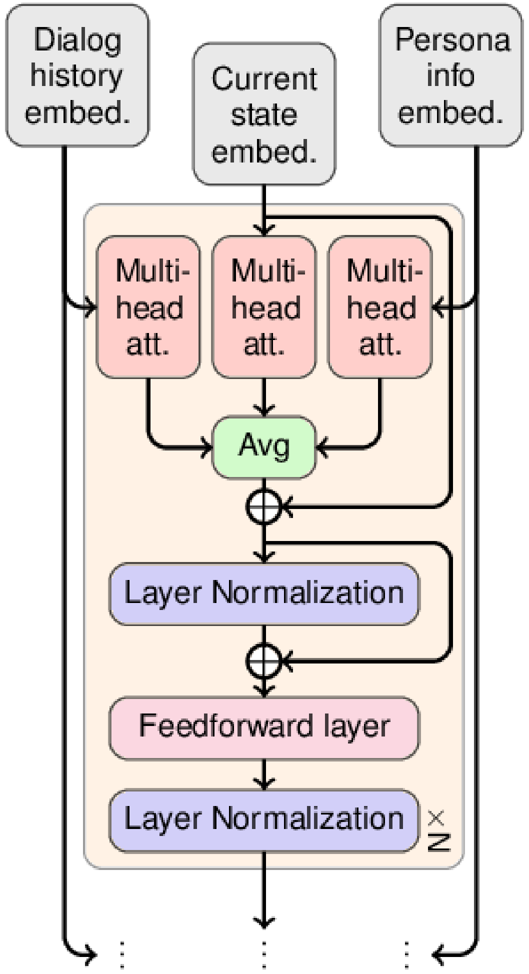
Layer Normalization

Feedforward Layer

Layer Normalization

N×

Output



Dialog history embed.

Current state embed.

Persona info embed.

Multi-head att.

Multi-head att.

Multi-head att.

Avg

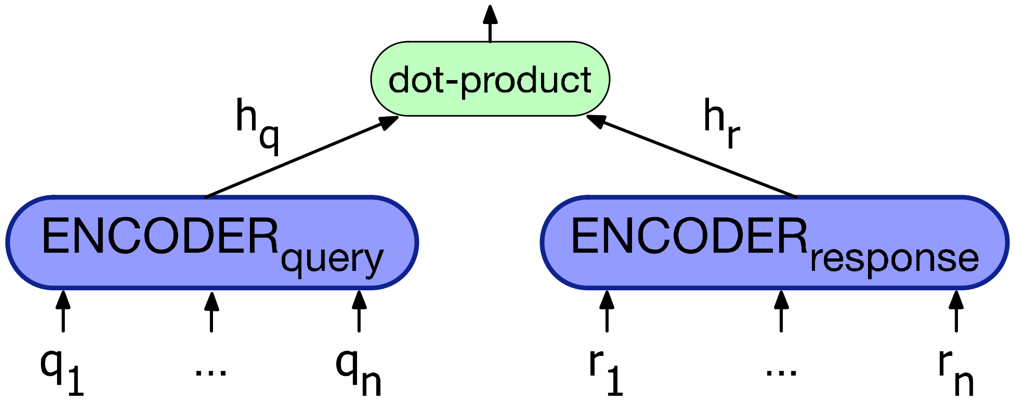
Layer Normalization

Feedforward layer

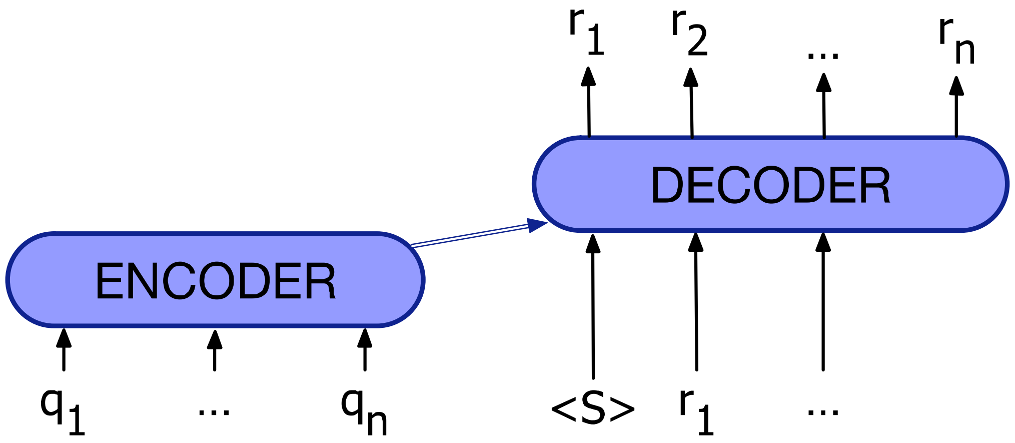
Layer Normalization

N×

Retrieval vs Generation


dot-product
h_q
h_r
$\text{ENCODER}_{\text{query}}$
$\text{ENCODER}_{\text{response}}$
q_1
…
q_n
r_1
…
r_n

(a) Response by Retrieval



(b) Response by Generation

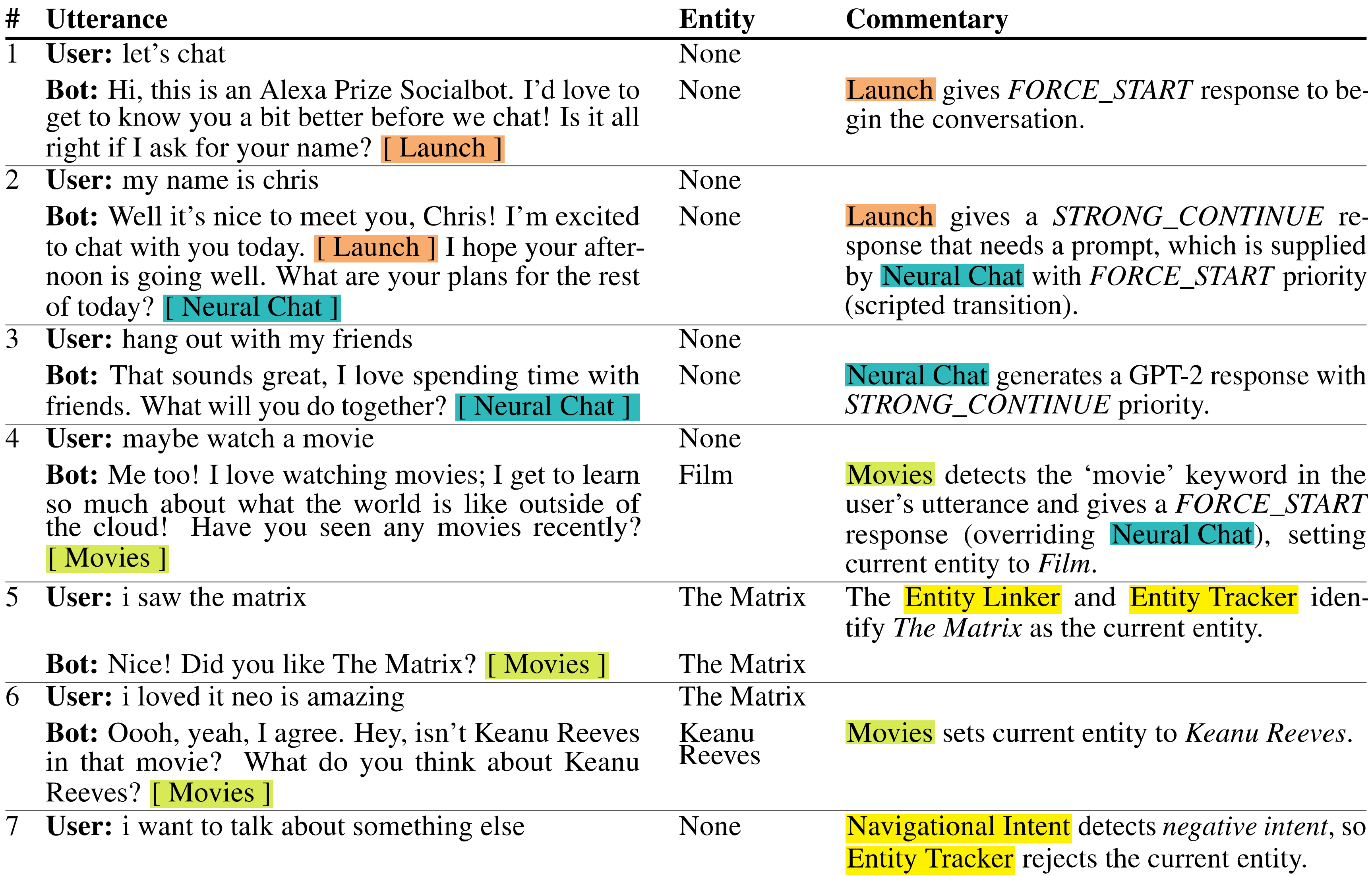
Response by Retrieving and Refining Knowledge

- 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.

Chirpy Cardinal



#	Utterance	Entity	Commentary
1	**User:** let's chat	None	
	Bot: 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]	None	Launch gives *FORCE_START* response to begin the conversation.
2	**User:** my name is chris	None	
	Bot: 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]	None	Launch gives a *STRONG_CONTINUE* response that needs a prompt, which is supplied by Neural Chat with *FORCE_START* priority (scripted transition).
3	**User:** hang out with my friends	None	
	Bot: That sounds great, I love spending time with friends. What will you do together? [Neural Chat]	None	Neural Chat generates a GPT-2 response with *STRONG_CONTINUE* priority.
4	**User:** maybe watch a movie	None	
	Bot: 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]	Film	Movies detects the 'movie' keyword in the user's utterance and gives a *FORCE_START* response (overriding Neural Chat), setting current entity to *Film*.
5	**User:** i saw the matrix	The Matrix	The Entity Linker and Entity Tracker identify *The Matrix* as the current entity.
	Bot: Nice! Did you like The Matrix? [Movies]	The Matrix	
6	**User:** i loved it neo is amazing	The Matrix	
	Bot: Oooh, yeah, I agree. Hey, isn't Keanu Reeves in that movie? What do you think about Keanu Reeves? [Movies]	Keanu Reeves	Movies sets current entity to *Keanu Reeves*.
7	**User:** i want to talk about something else	None	Navigational Intent detects *negative intent*, so Entity Tracker rejects the current entity.

(Paranjape et al. 2020)

Chatbots: pro and con

- 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 Step

Integrating chatbot ability into frame-based agents

Outline

- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
- **Task-based Chatbots**
  - GUS
  - DSA
- Evaluation

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
  - Buying a product

Architecture:

- First proposed of a GUS system in 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?
DEP TIME	time	"What time would you like to leave?
AIRLINE	line	"What is your preferred airline?

The Frame

- Types in GUS, as in modern frame-based dialogue agents, have hierarchical structure
    - e.g., the date type in GUS is itself a frame with slots with types like integer or members of sets of weekday names:

```
DATE

MONTH:NAME

YEAR:INTEGER

DAY:(BOUNDED-INTEGER 1 31)

WEEKDAY:(MEMBER (Sunday Monday Tuesday Wednesday Thursday Friday Saturday))
```

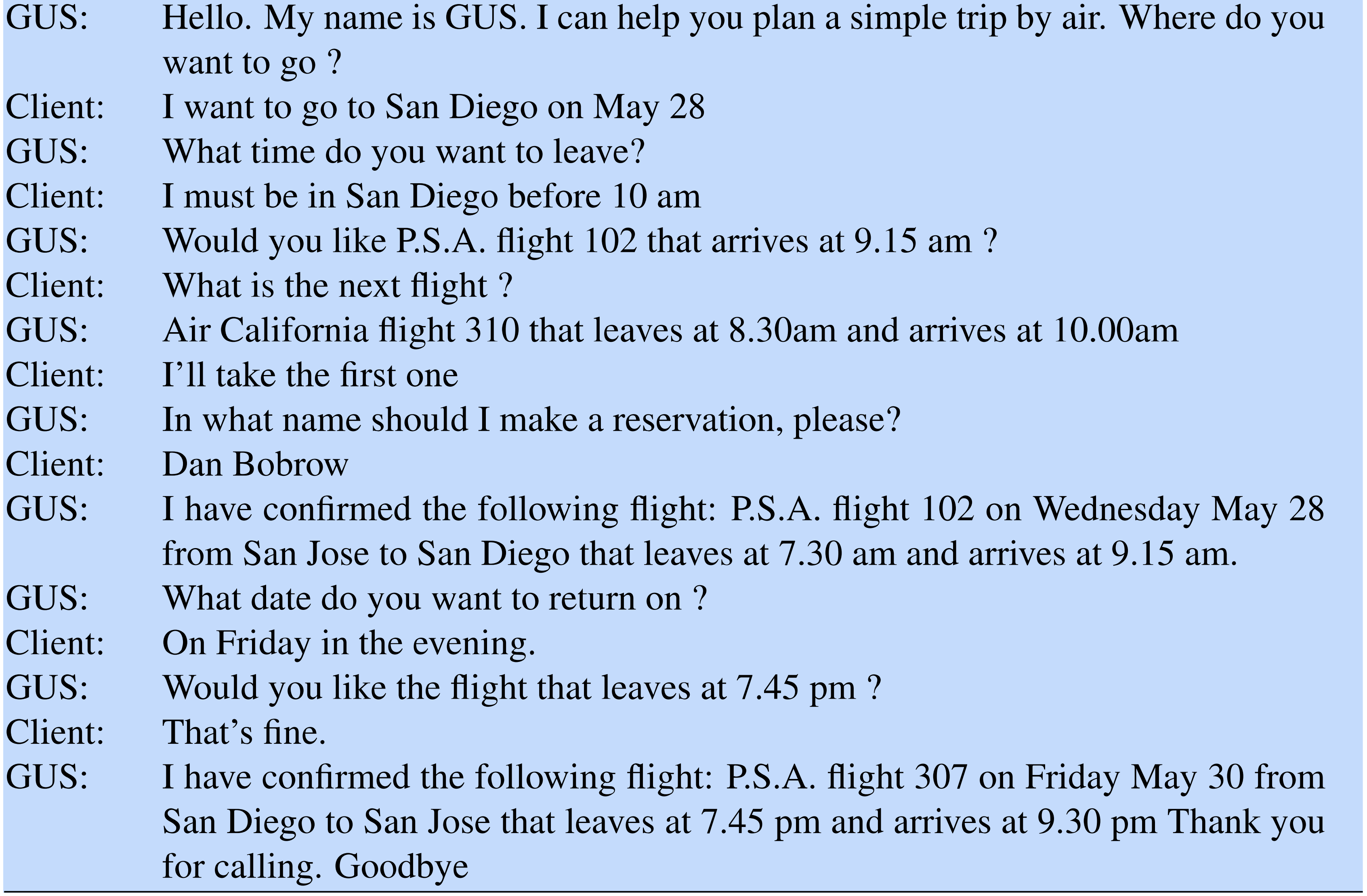
Two Basic Architectures

- The **GUS** architecture
  - Simple frame-based dialogue system
  - Sometimes just called "**frame-based**" architecture
  - Over 40 years old, but still used in most industrial task-based dialogue agents

*Bobrow, Daniel G., Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. 1977. "**GUS**, a frame-driven dialog system." Artificial Intelligence 8, 2:155-173.*

- The **dialogue-state** architecture
  - Extension of GUS
  - More common in research systems
  - Some aspects making their way into industrial systems

Frame-based Dialogue Agents



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

Outline

- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
- **Task-based Chatbots**
  - **GUS**
  - DSA
- Evaluation

GUS for Task-based Chatbots

- **Control Structure**
- NLU for Filling Slots in GUS
- Other Components of Frame-based Dialogue

Control Structure for GUS Frame Architecture

- System asks questions of user, filling any slots that user specifies

- User might fill many slots at a time:

I want a flight from San Francisco to Denver one way leaving after five p.m. on Tuesday.

- When frame is filled, do database query

GUS slots have condition-action rules attached

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

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

GUS for Task-based Chatbots

- Control Structure
- **NLU for Filling Slots in GUS**
- Other Components of Frame-based Dialogue

NLU for Filling Dialog Slots

1. **Domain classification**
    - Asking weather? Booking a flight? Programming alarm clock?

2. **Intent Determination**
    - Find a Movie, Show Flight, Remove Calendar Appointment

3. **Slot Filling**
    - Extract the actual slots and fillers

NLU for Filling Dialog 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 Dialog Slots

`Wake me tomorrow at six.`

```
DOMAIN:   ALARM-CLOCK
INTENT:   SET-ALARM
```

```
TIME:    2017-07-01 0600
```

How to fill slots?
(1) Rule-based Slot Filling

- Write regular expressions or grammar rules

```
Wake me (up) | set (the|an) alarm | get me up
```

- Do text normalization

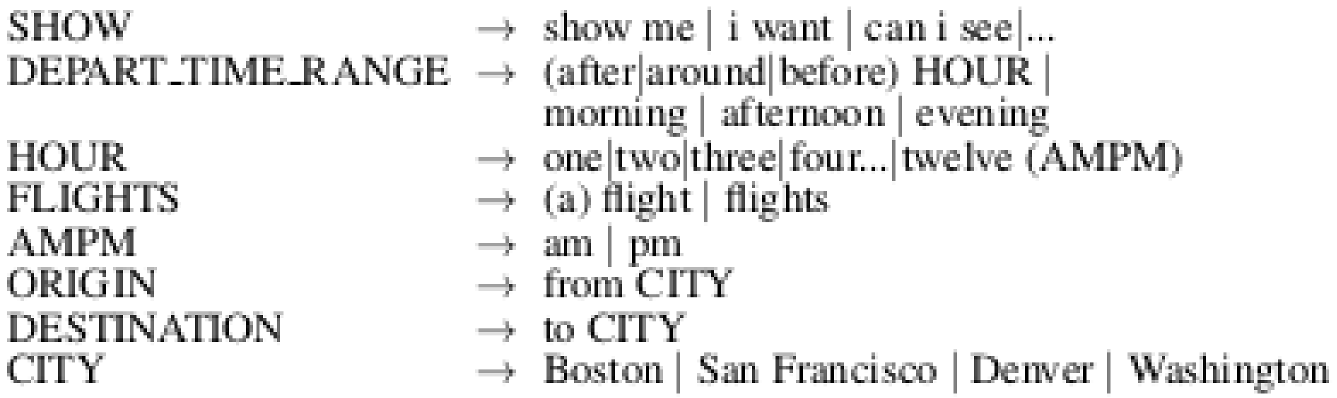
Rule-based Slot Filling

- Rule-based research systems like the Phoenix system (Ward and Issar, 1994) consist of large hand-designed semantic grammars with thousands of rules

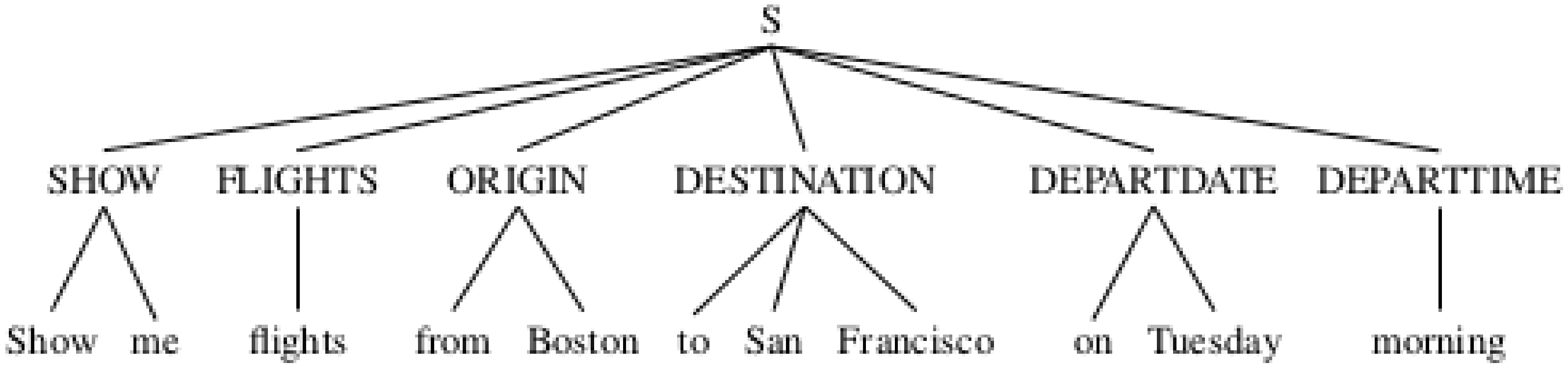
- A semantic grammar is a context-free grammar in which the left-hand side of each rule corresponds to the semantic entities being expressed (i.e., the slot names)

- Semantic grammars can be parsed by any CFG parsing algorithm, resulting in a hierarchical labeling of the input string with semantic node labels

Rule-based Slot Filling



SHOW → show me | i want | can i see|...
DEPART_TIME_RANGE → (after|around|before) HOUR |
morning | afternoon | evening
HOUR → one|two|three|four...|twelve (AMPM)
FLIGHTS → (a) flight | flights
AMPM → am | pm
ORIGIN → from CITY
DESTINATION → to CITY
CITY → Boston | San Francisco | Denver | Washington



GUS

- Many industrial dialogue systems employ the GUS architecture
- But use supervised machine learning for slot-filling instead of these kinds of rules
- We will discuss later

GUS for Task-based Chatbots

- Control Structure
- NLU for Filling Slots in GUS
- **Other Components of Frame-based Dialogue**

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

- A 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

Outline

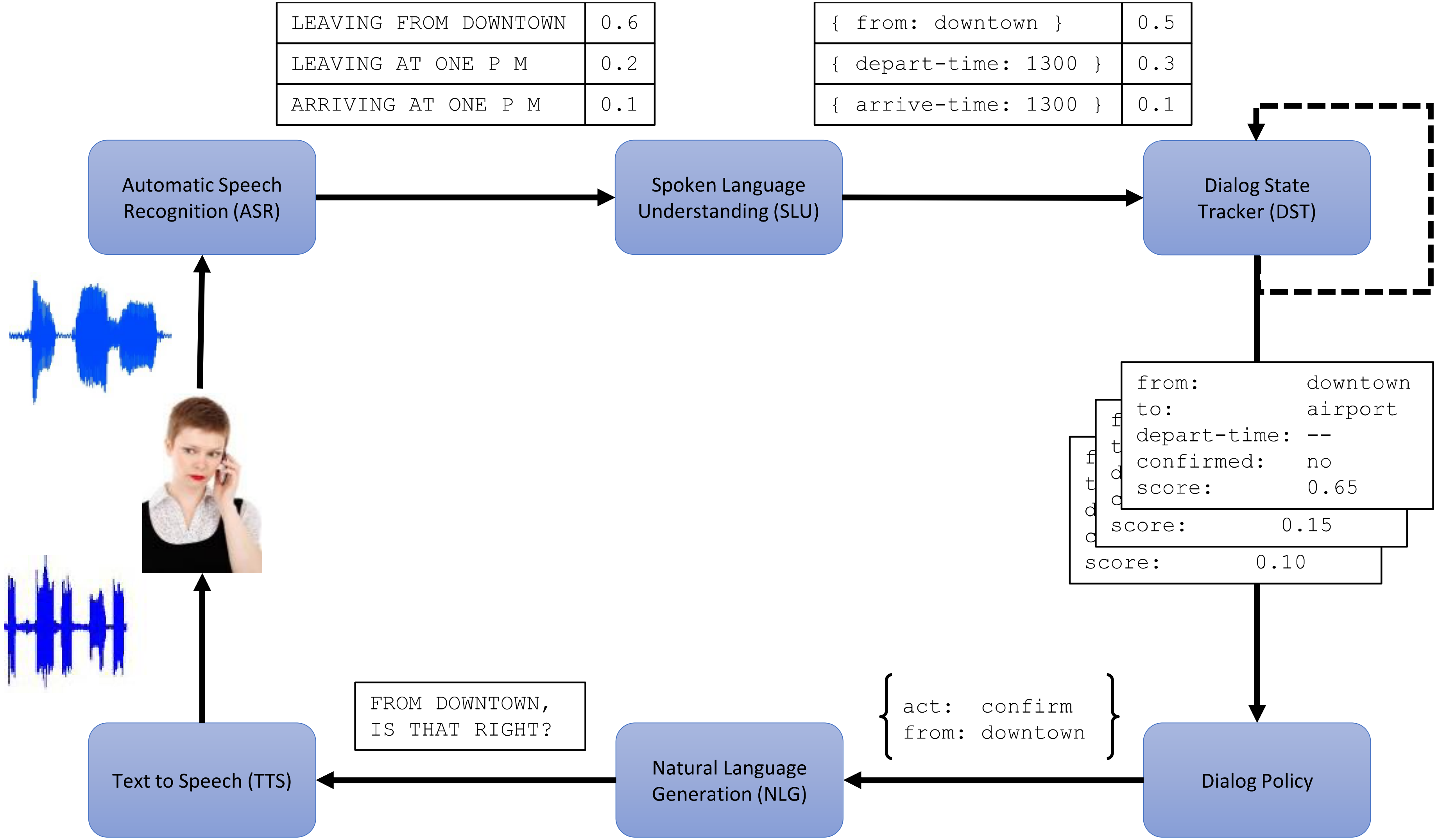
- **Introduction**
- Properties of Human Conversation
- Conversational Chatbots
- **Task-based Chatbots**
    - GUS
    - **DSA**
- Evaluation

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



LEAVING FROM DOWNTOWN	0.6
LEAVING AT ONE P M	0.2
ARRIVING AT ONE P M	0.1

{ from: downtown }	0.5
{ depart-time: 1300 }	0.3
{ arrive-time: 1300 }	0.1

Automatic Speech Recognition (ASR)
Spoken Language Understanding (SLU)
Dialog State Tracker (DST)
Dialog Policy
Natural Language Generation (NLG)
Text to Speech (TTS)

```
from:          downtown
to:            airport
depart-time:   --
confirmed:     no
score:         0.65
```

```
score:         0.15
```

```
score:         0.10
```

```
{ act:   confirm
  from:  downtown }
```

```
FROM DOWNTOWN,
IS THAT RIGHT?
```

Williams, Jason D., Antoine Raux, and Matthew Henderson. "The dialog state tracking challenge series: A review." Dialogue & Discourse 7, no. 3 (2016): 4-33.

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-State Architecture vs GUS

- Slot fillers use ML rather than rules

- Dialogue policy can help a system decide
  - When to answer the user’s questions
  - When to instead ask the user a clarification question
  - When to make a suggestion
  - …

- Dialogue state systems have a natural language generation component
  - In GUS, the sentences that the generator produced were all from pre-written templates
  - But a more sophisticated generation component can condition on the exact context to produce turns that seem much more natural

Dialogue-state Architecture

- **Dialogue Acts**
- Slot Filling
- Dialogue State Tracking
- Dialogue Policy
- NLG in the Dialogue-state Model

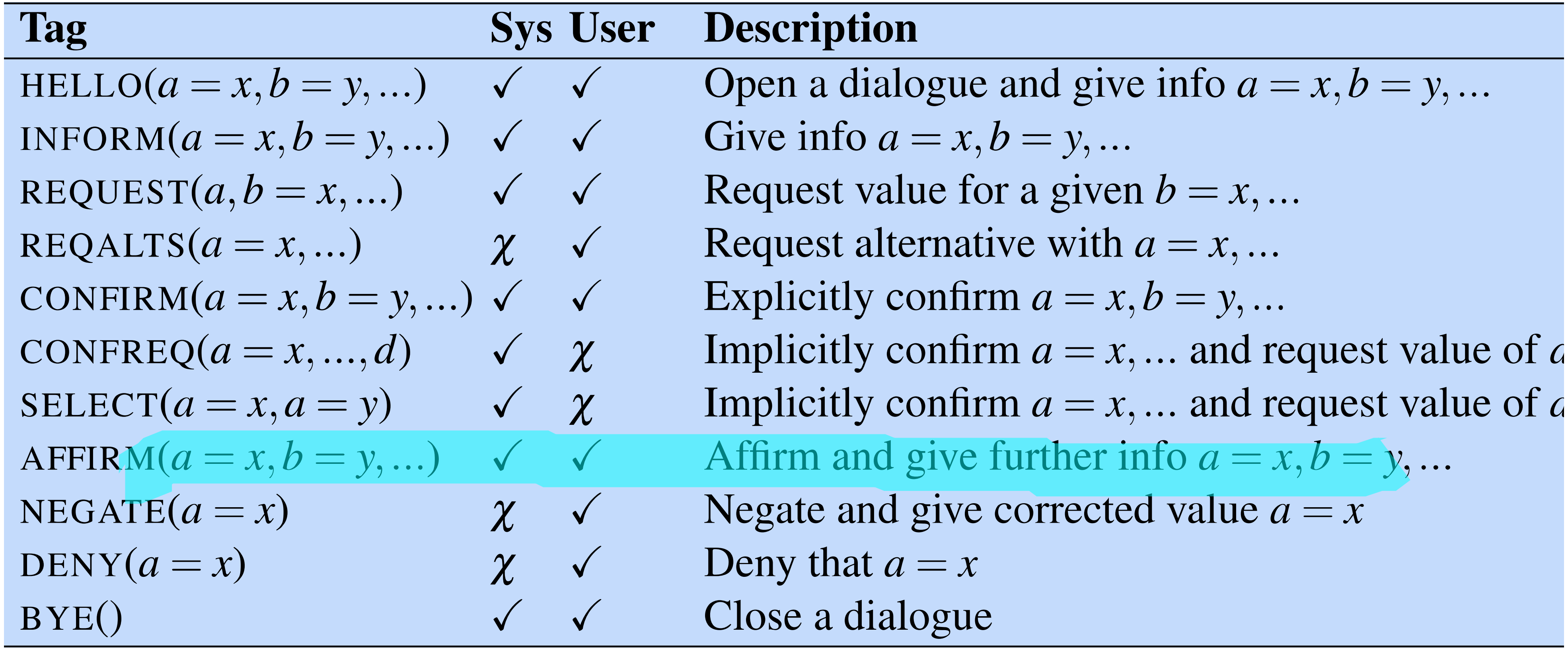
Dialogue Acts

- Dialogue acts represent the interactive function of the turn or sentence
- Combine the ideas of **speech acts** and **grounding** into a single representation

- Different types of dialogue systems require labeling different kinds of acts
  - The tagset —defining what a dialogue act is exactly— tends to be designed for particular tasks

Dialogue Acts

- Sample Tagset

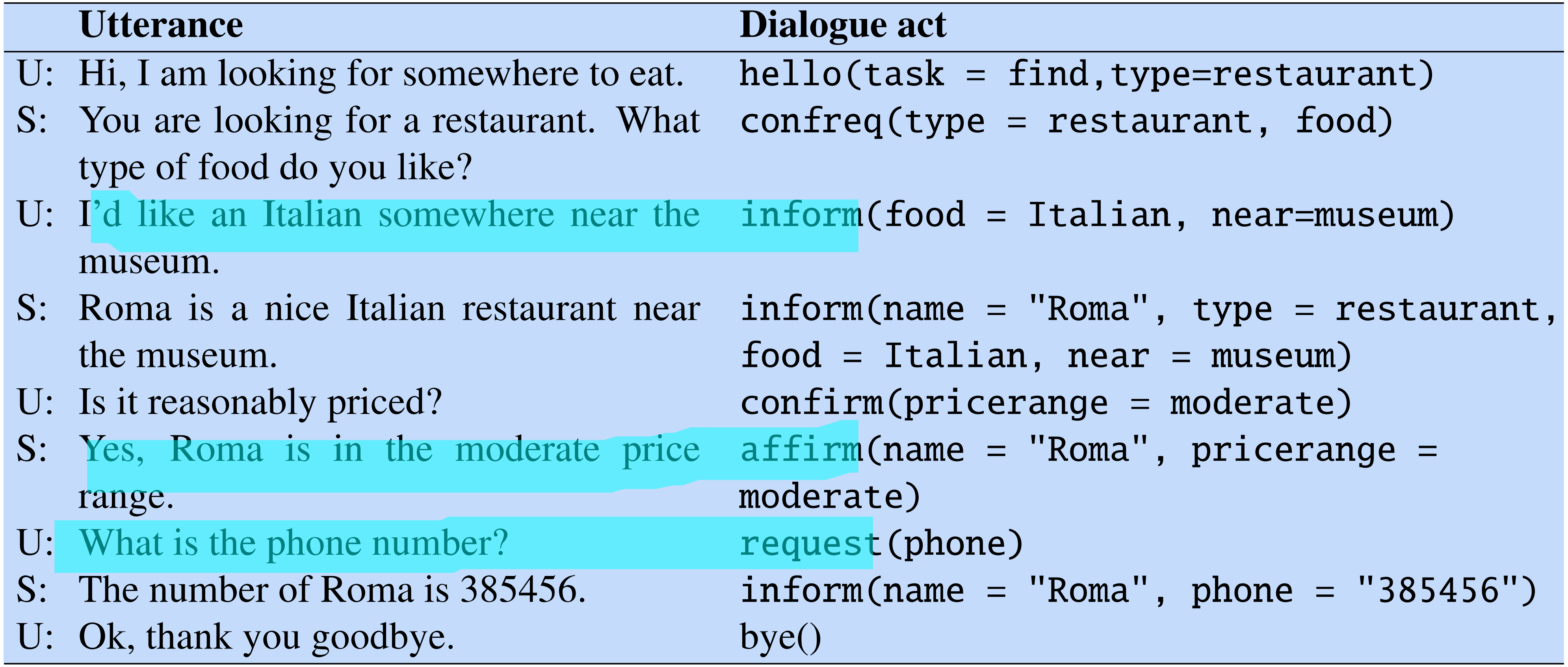


Tag	Sys	User	Description
HELLO$(a=x, b=y, \ldots)$	✓	✓	Open a dialogue and give info $a=x, b=y, \ldots$
INFORM$(a=x, b=y, \ldots)$	✓	✓	Give info $a=x, b=y, \ldots$
REQUEST$(a, b=x, \ldots)$	✓	✓	Request value for a given $b=x, \ldots$
REQALTS$(a=x, \ldots)$	χ	✓	Request alternative with $a=x, \ldots$
CONFIRM$(a=x,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

(Young et al., 2010)

Dialogue Acts

- Sample data (The content of each dialogue acts communicates with the slot fillers)



	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()

(Young et al., 2010)

Dialogue-state Architecture

- Dialogue Acts
- **Slot Filling**
- Dialogue State Tracking
- Dialogue Policy
- NLG in the Dialogue-state Model

Slot Filling: Machine learning

- Machine learning classifiers to map words to semantic frame-fillers
- Given a set of labeled sentences

Input: `"I want to fly to San Francisco on Monday afternoon please"`

Output: `Destination: SF`
`Depart-time: Monday afternoon`

- Build a classifier to map from one to the other
- Requirements: Lots of labeled data

Slot Filling as Sequence Labeling

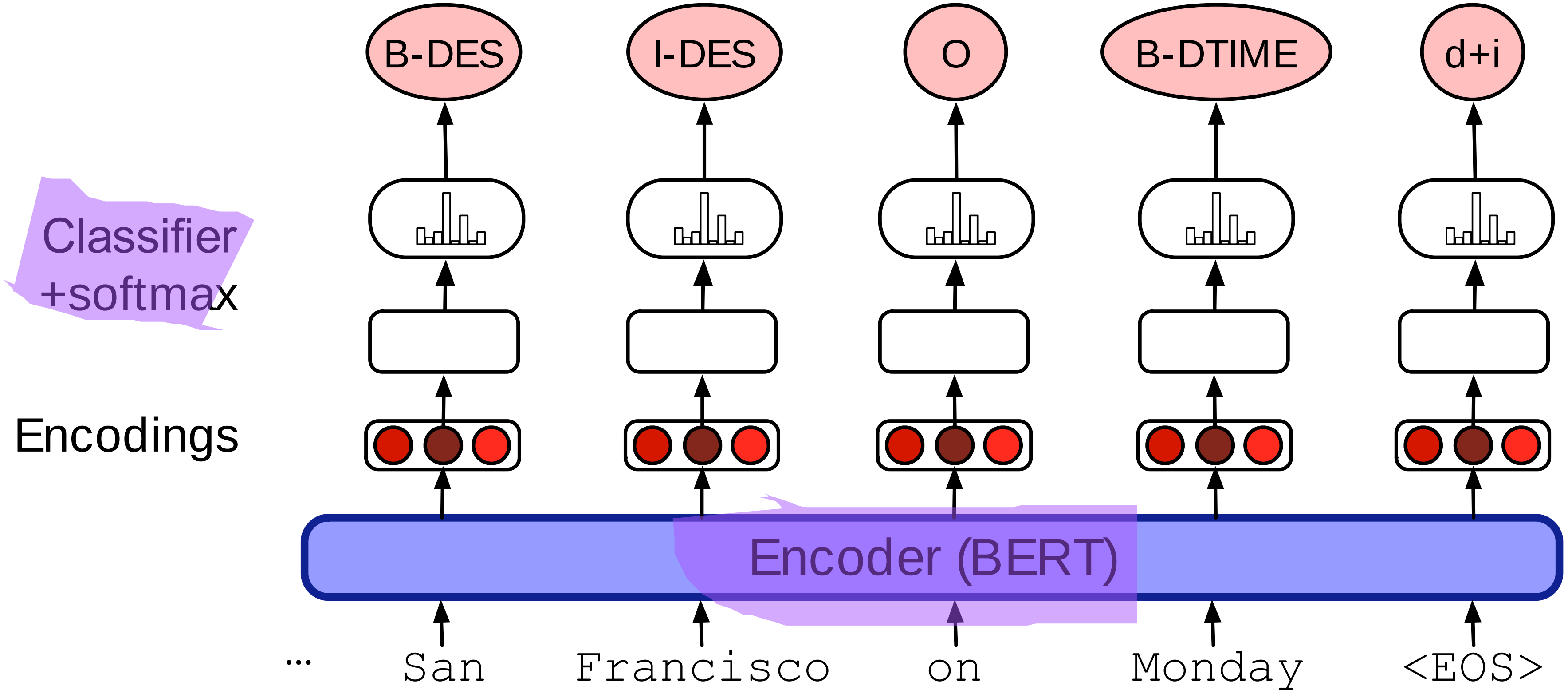
- The **BIO tagging** paradigm
- Idea: Train a classifier to label each input word with a tag that tells us what slot (if any) it fills

O	O	O	O	O	B-DES	I-DES	O	B-DEPTIME	I-DEPTIME	O
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

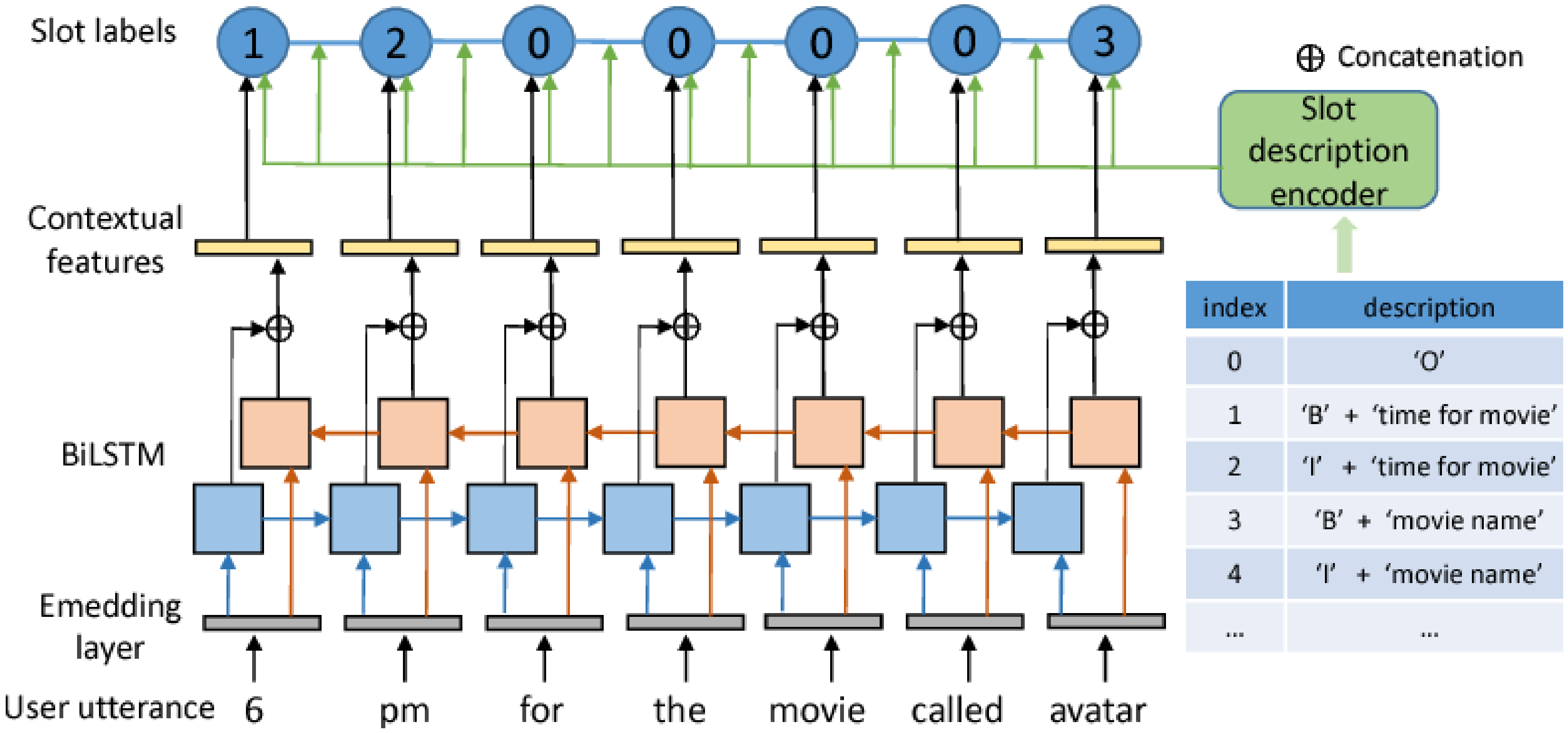
Slot Filling Using Contextual Embeddings

- Can do domain and intent too:
  - e.g., generate the label "AIRLINE_TRAVEL + SEARCH_FLIGHT"



Slot Filling Using RNNs

- Using BiLSTM



index	description
0	‘O’
1	‘B’ + ‘time for movie’
2	‘I’ + ‘time for movie’
3	‘B’ + ‘movie name’
4	‘I’ + ‘movie name’
…	…

Dialogue-state Architecture

- Dialogue Acts
- Slot Filling
- **Dialogue State Tracking**
- Dialogue Policy
- NLG in the Dialogue-state Model

Dialogue State Tracking

- The job of the dialogue-state tracker is to determine
  - The current state of the frame (the fillers of each slot)
  - As well as the user’s most recent dialogue act

- Dialogue-state includes more than just the slot-fillers expressed in the current sentence

- It includes the entire state of the frame at this point, summarizing all of the user’s constraints

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

Dialogue State Tracking

- I'd like Cantonese food near the Mission district.

→ `inform(food=cantonese, area=mission).`

- Dialogue act interpretation use a supervised classification algorithm to choose `inform`
  - Used handlabeled dialog acts to predicting the dialogue act tag
  - Based on Encodings of current sentence + prior dialogue acts

Simple and Complex Dialogue State Tracking

- Simple dialogue state tracker:
  - Run a slot-filler after each sentence

- More complex model:
  - Can make use of the reading-comprehension architectures
  - e.g., the model of Gao et al. (2019) trains a classifier for each slot to decide whether its value is being changed in the current sentence or should be carried over from the previous sentences
  - If the slot value is being changed, a span-prediction model is used to predict the start and end of the span with the slot filler

Detecting Correction Acts

- An special case of dialogue act detection
- If system misrecognizes an utterance
- User might make a **correction**
  - Repeat themselves
  - Rephrasing
  - Saying “no” to a confirmation question

Detecting Correction Acts

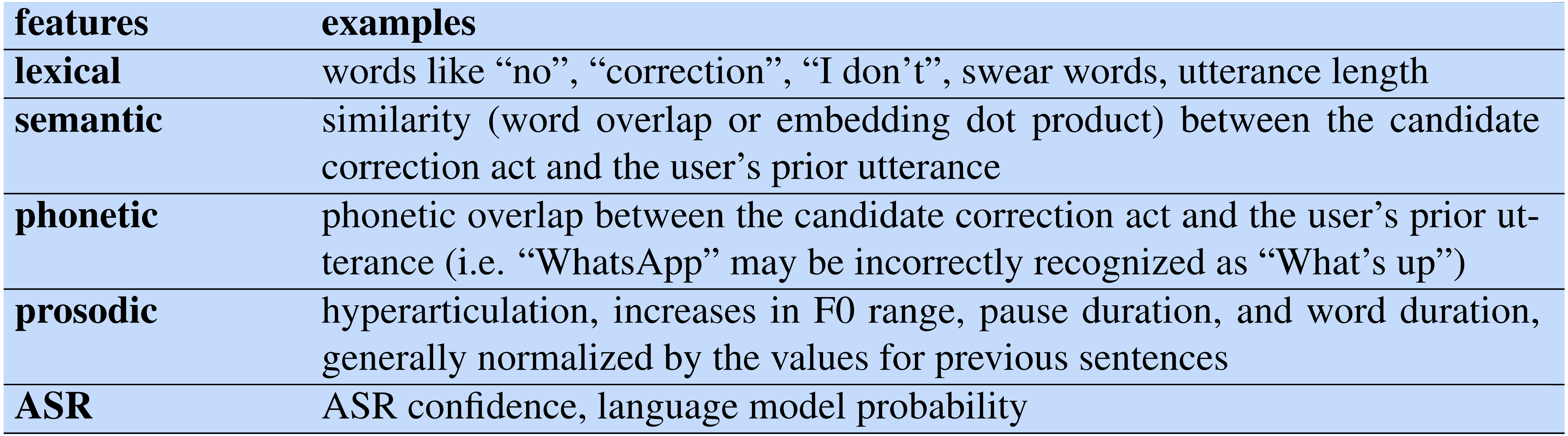
- Corrections are harder to recognize!
- From speech, corrections are misrecognized twice as often (in terms of word error rate) as non-corrections! (Swerts et al 2000)
- Hyperarticulation (exaggerated prosody) is a large factor:
  - "I said BAL-TI-MORE, not Boston"

Detecting Correction Acts

- User corrections tend to be either exact repetitions or repetitions with one or more words omitted, although they may also be paraphrases of the original utterance
- Detecting reformulations or correction acts can be part of the general dialogue act detection classifier
- Alternatively, because the cues to these acts tend to appear in different ways than for simple acts (like INFORM or REQUEST), we can make use of features orthogonal to simple contextual embedding features

Features for Detecting Corrections in Spoken Dialogue

- Some typical features



features	examples
lexical	words like “no”, “correction”, “I don’t”, swear words, utterance length
semantic	similarity (word overlap or embedding dot product) between the candidate correction act and the user’s prior utterance
phonetic	phonetic overlap between the candidate correction act and the user’s prior utterance (i.e. “WhatsApp” may be incorrectly recognized as “What’s up”)
prosodic	hyperarticulation, increases in F0 range, pause duration, and word duration, generally normalized by the values for previous sentences
ASR	ASR confidence, language model probability

Dialogue-state Architecture

- Dialogue Acts
- Slot Filling
- Dialogue State Tracking
- **Dialogue Policy**
- NLG in the Dialogue-state Model

Dialogue Policy

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

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

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

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

Dialogue Policy with Reinforcement Learning

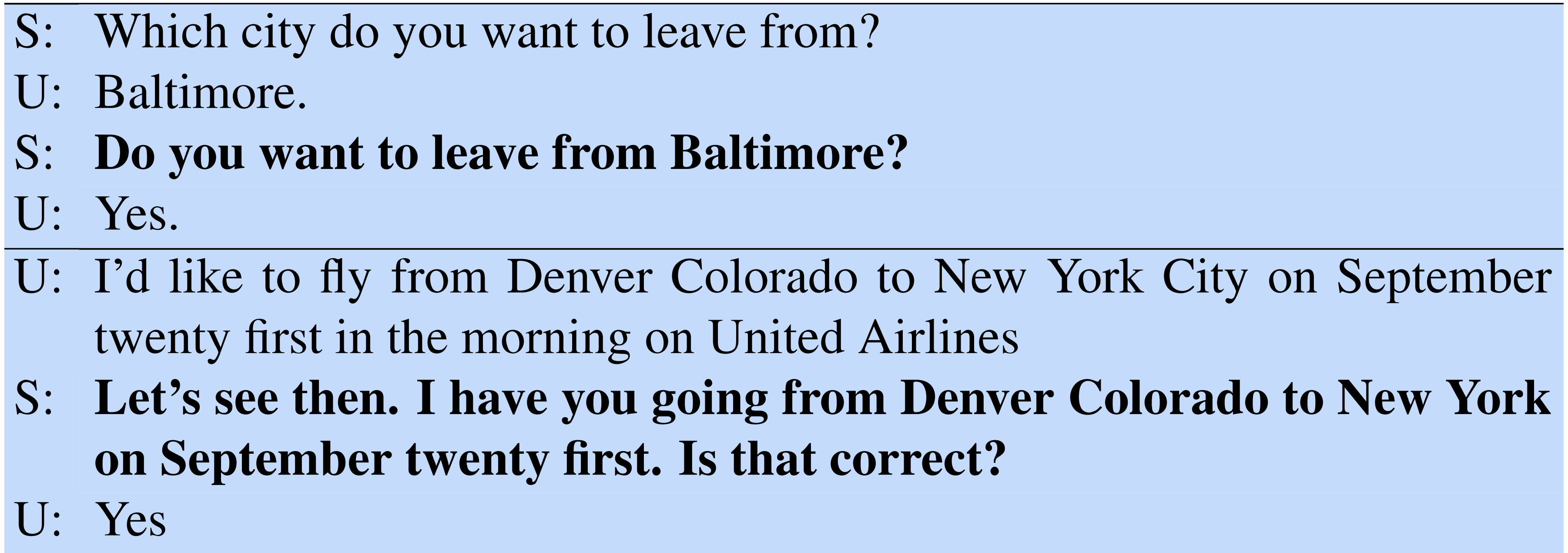
- To decide which action to take, a reinforcement learning system gets a reward at the end of the dialogue, and uses that reward to train a policy to take actions
- Example: a movie-recommendation dialogue system (Fazel-Zarandi et al. 2017)
  - The action space has only three actions: EXECUTE, CONFIRM, and ELICIT
    - **EXECUTE** which sends a query to the database and answers the user's question
    - **CONFIRM** which clarifies the intent or slot with the users (e.g., "Do you want movies directed by Christopher Nolan?")
    - **ELICIT** which asks the user for missing information (e.g., "Which movie are you talking about?")
  - The system rewards includes:
    - A large positive reward if the dialogue system terminates with the correct slot representation at the end
    - A large negative reward if the slots are wrong
    - A small negative reward for confirmation and elicitation questions to keep the system from re-confirming everything

Policy Example: Confirmation and Rejection

- Dialogue systems make errors
- So they need 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

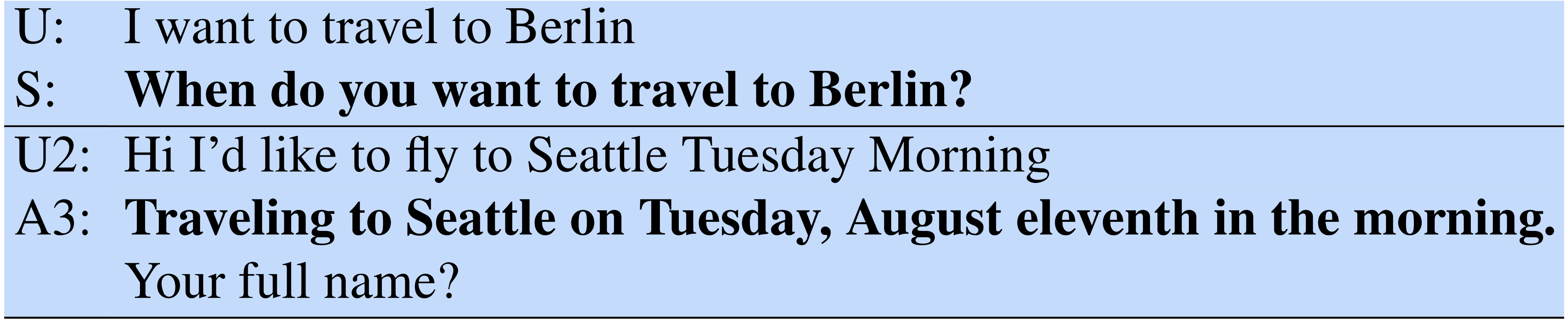
Explicit Confirmation Strategy



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

Implicit Confirmation Strategy



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?

Confirmation Strategy Tradeoffs

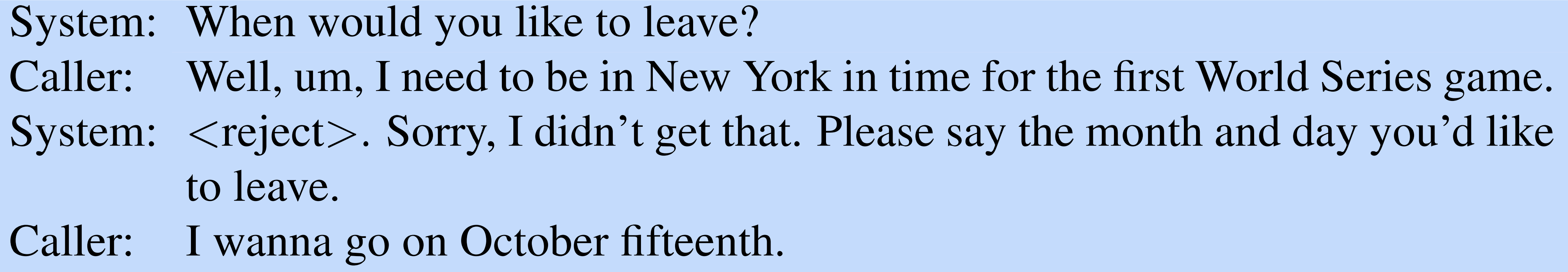
- Explicit confirmation makes it easier for users to correct the system's misrecognitions since a user can just answer "no" to the confirmation question
- But explicit confirmation is also awkward and increases the length of the conversation (Danieli and Gerbino 1995, Walker et al. 1998)
- The explicit confirmation dialogue fragments above sound non-natural and definitely non-human
- Implicit confirmation is much more conversationally natural

Rejection

I’m sorry, I didn’t understand that.

Progressive Prompting for Rejection

- Also known as escalating detail
- Give user guidance about what they can say
    - Don't just repeat the question "When would you like to leave?"



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.

Rapid Reprompting

- The system rejects an utterance just by saying
  - "I'm sorry?"
  - "What was that?"

- Only if the caller's utterance is rejected a second time, the system starts applying progressive prompting

- Experiments showed that users greatly prefer rapid reprompting as a first-level error prompt (Cohen et al., 2004)

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 implictly
$\geq \gamma$	very high confidence	don’t confirm at all

Dialogue-state Architecture

- Dialogue Acts
- Slot Filling
- Dialogue State Tracking
- Dialogue Policy
- **NLG in the Dialogue-state Model**

Natural Language Generation

- NLG in information-state architecture modeled in two stages:
    - **Content planning** (what to say)
    - **Sentence realization** (how to say it)

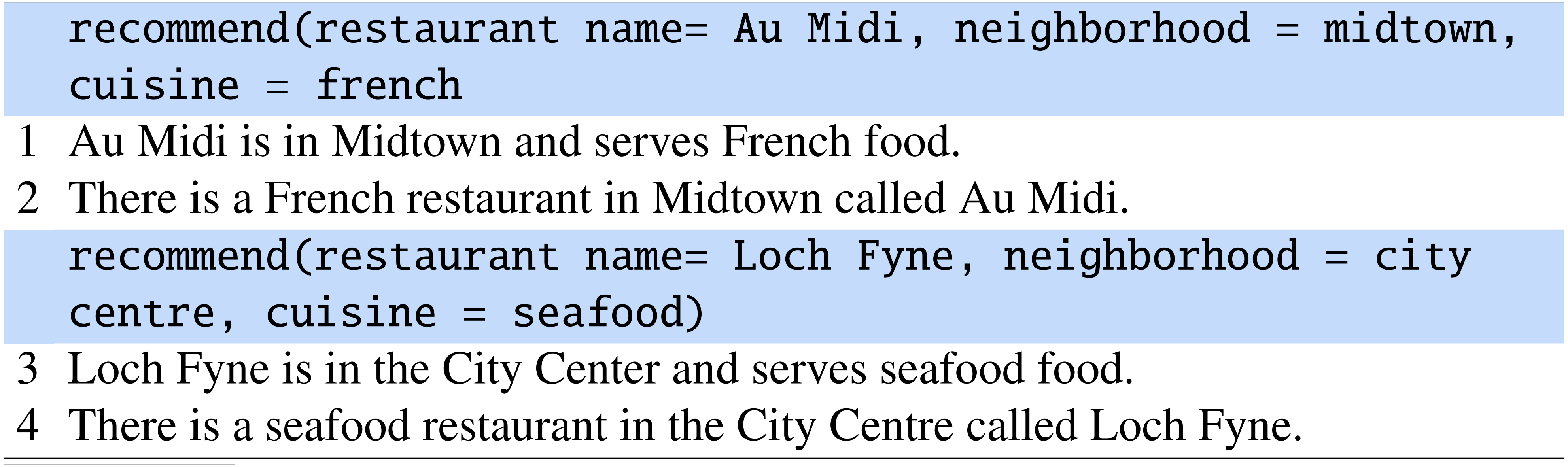
- Content planning is normally done by the dialogue policy
- We'll focus on sentence realization here

Sentence Realization

- Assume the dialogue policy performed content planning by choosing
  - The dialogue act to generate
  - 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)

Sentence Realization

- 2 examples



`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`

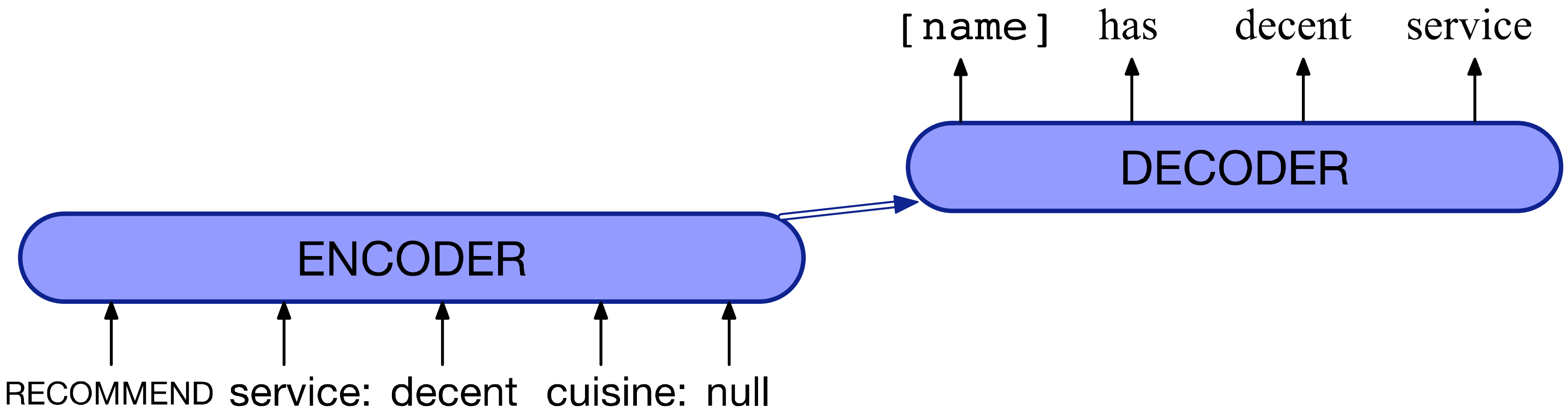
1 Au Midi is in Midtown and serves French food.
2 There is a French restaurant in Midtown called Au Midi.

`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`

NLG for Specific Dialogue Acts

- Designing NLG algorithms that are specific to a particular dialogue act
  - e.g., generating clarification questions

- In cases where the speech recognition fails to understand some part of the user’s utterance
  - Instead of generating a generic dialogue act REJECT use targeted clarification questions that reprise elements of the misunderstanding
    - (“Please repeat”, or “I don’t understand what you said”)

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

Outline

- Introduction
- Properties of Human Conversation
- Conversational Chatbots
- Task-based Chatbots
- **Evaluation**

Evaluating Chatbots and Task-based Dialogue

- Task-based dialogue:
  - Mainly by measuring task performance
- Conversational chatbots:
  - Mainly by human evaluation

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

Participant Evaluation

- **Avoiding Repetition:** How repetitive was this user?
  - Repeated themselves over and over
  - Sometimes said the same thing twice
  - Always said something new

Participant Evaluation

- **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

Participant Evaluation

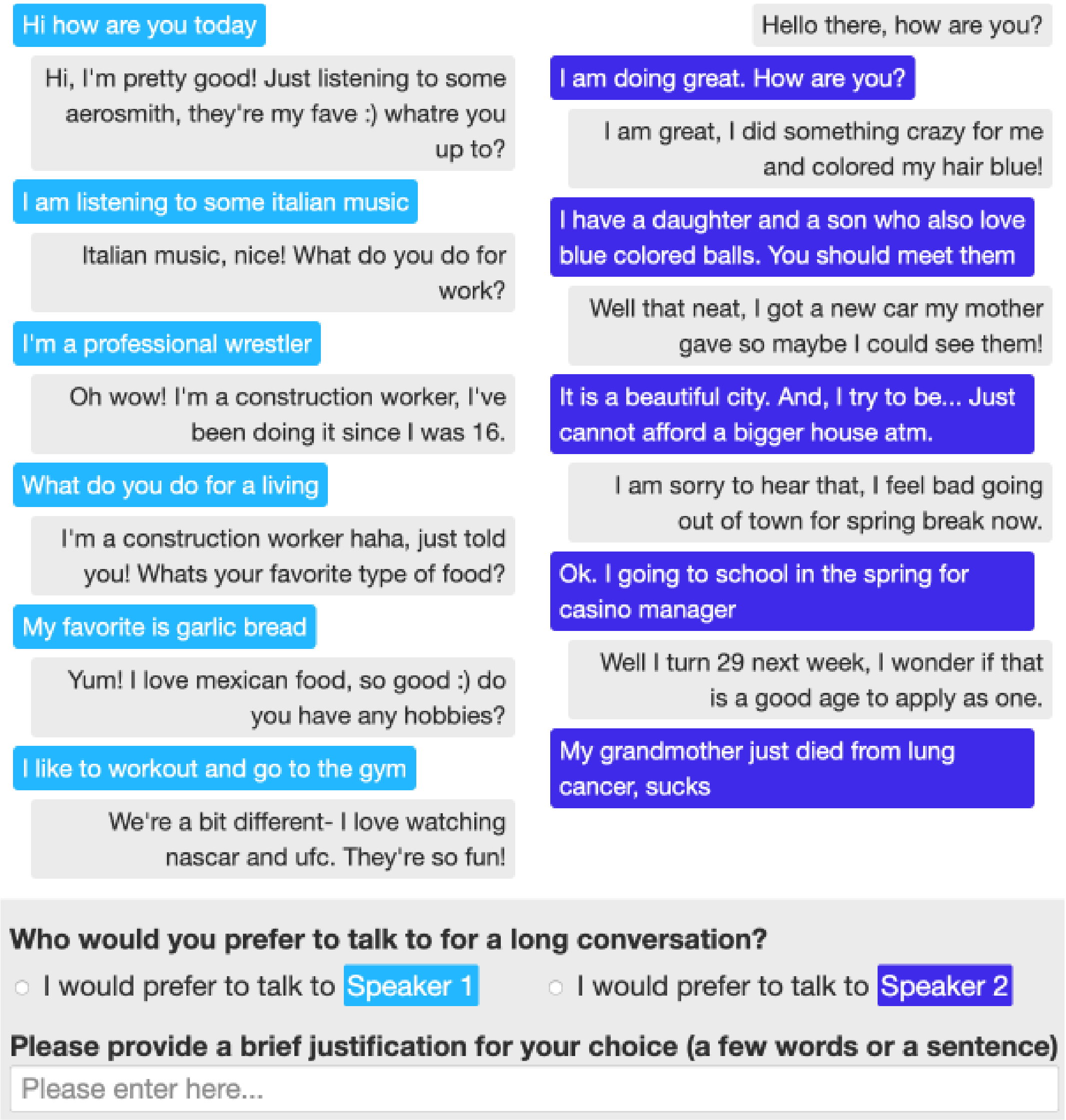
- **Engagingness:** How much did you enjoy talking to this user?
  - Not at all
  - A little
  - Somewhat
  - A lot

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?

Observer Evaluation: acute-eval

- The ACUTE-EVAL method (Li et el., 2019)



Hi how are you today

Hi, I'm pretty good! Just listening to some aerosmith, they're my fave :) whatre you up to?

I am listening to some italian music

Italian music, nice! What do you do for work?

I'm a professional wrestler

Oh wow! I'm a construction worker, I've been doing it since I was 16.

What do you do for a living

I'm a construction worker haha, just told you! Whats your favorite type of food?

My favorite is garlic bread

Yum! I love mexican food, so good :) do you have any hobbies?

I like to workout and go to the gym

We're a bit different- I love watching nascar and ufc. They're so fun!

Hello there, how are you?

I am doing great. How are you?

I am great, I did something crazy for me and colored my hair blue!

I have a daughter and a son who also love blue colored balls. You should meet them

Well that neat, I got a new car my mother gave so maybe I could see them!

It is a beautiful city. And, I try to be... Just cannot afford a bigger house atm.

I am sorry to hear that, I feel bad going out of town for spring break now.

Ok. I going to school in the spring for casino manager

Well I turn 29 next week, I wonder if that is a good age to apply as one.

My grandmother just died from lung cancer, sucks

Who would you prefer to talk to for a long conversation?

○ I would prefer to talk to Speaker 1 ○ I would prefer to talk to Speaker 2

Please provide a brief justification for your choice (a few words or a sentence)

Please enter here...

Automatic Evaluation

- Automatic evaluation metrics are used for comparing machine generated text with gold texts, like
  - BLEU (mainly used for machine translation)
  - ROUGE (mainly used for text summarization)

- Automatic evaluation metrics are generally not used for chatbots
  - They are so many possible responses for a given turn
  - They correlate poorly with human judgements

Adversarial Evaluation

- One current research direction toward automatic evaluation: **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

Evaluating Chatbots and Task-based Dialogue

- Task-based dialogue:
  - Mainly by measuring task performance
- Conversational chatbots:
  - Mainly by human evaluation

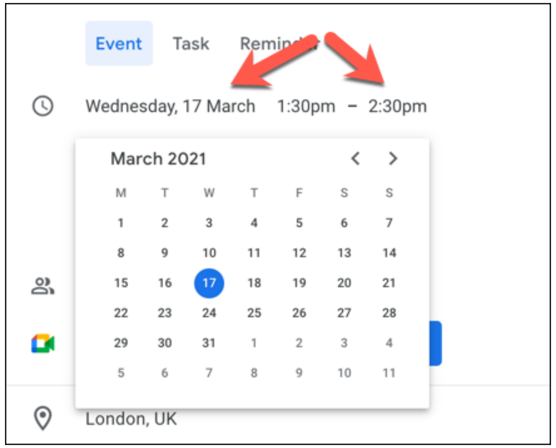
Task-based Evaluation

- End-to-end evaluation
  - Measure absolute task success
  - Measure user satisfaction
- Slot Error Rate for a Sentence

Evaluating by Task Success!

- Measure absolute task success
  - Did the system book the right plane flight?

  - Did the system put the right event on the calendar?



Wednesday, 17 March 1:30pm – 2:30pm

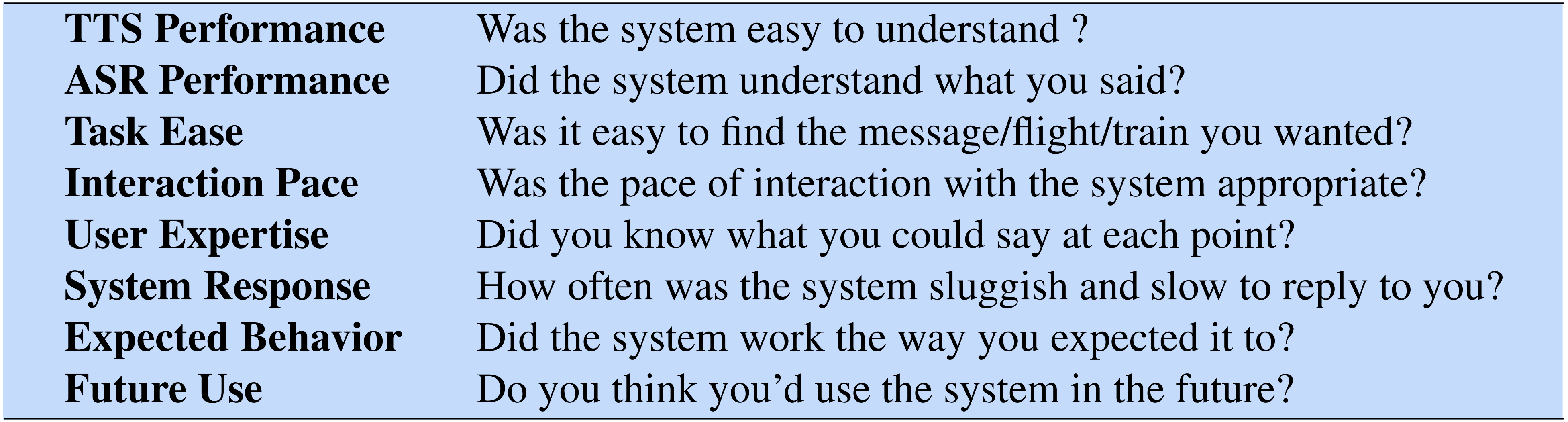
March 2021

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

London, UK

Evaluating by User Satisfaction

- More Fine-grained Metrics
- A User Satisfaction Survey:



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 message/flight/train 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	How often was the system 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?

Evaluating Slot Filling

- Slot Error Rate for a Sentence

$$\frac{\text{\# of inserted/deleted/subsituted slots}}{\text{\# of total reference slots for sentence}}$$

- Precision, Recall, F-score for slot filling evaluation

Evaluation Metrics: Slot Error Rate

"Make an appointment with Chris at 10:30 in Gates 104"

Slot	**Filler**
PERSON	Chris
TIME	11:30 a.m.
ROOM	Gates 104

Slot error rate: 1/3

Task success: At end, was the correct meeting added to the calendar?

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

Further Reading

- Speech and Language Processing (3rd ed. draft)
  - Chapter 15