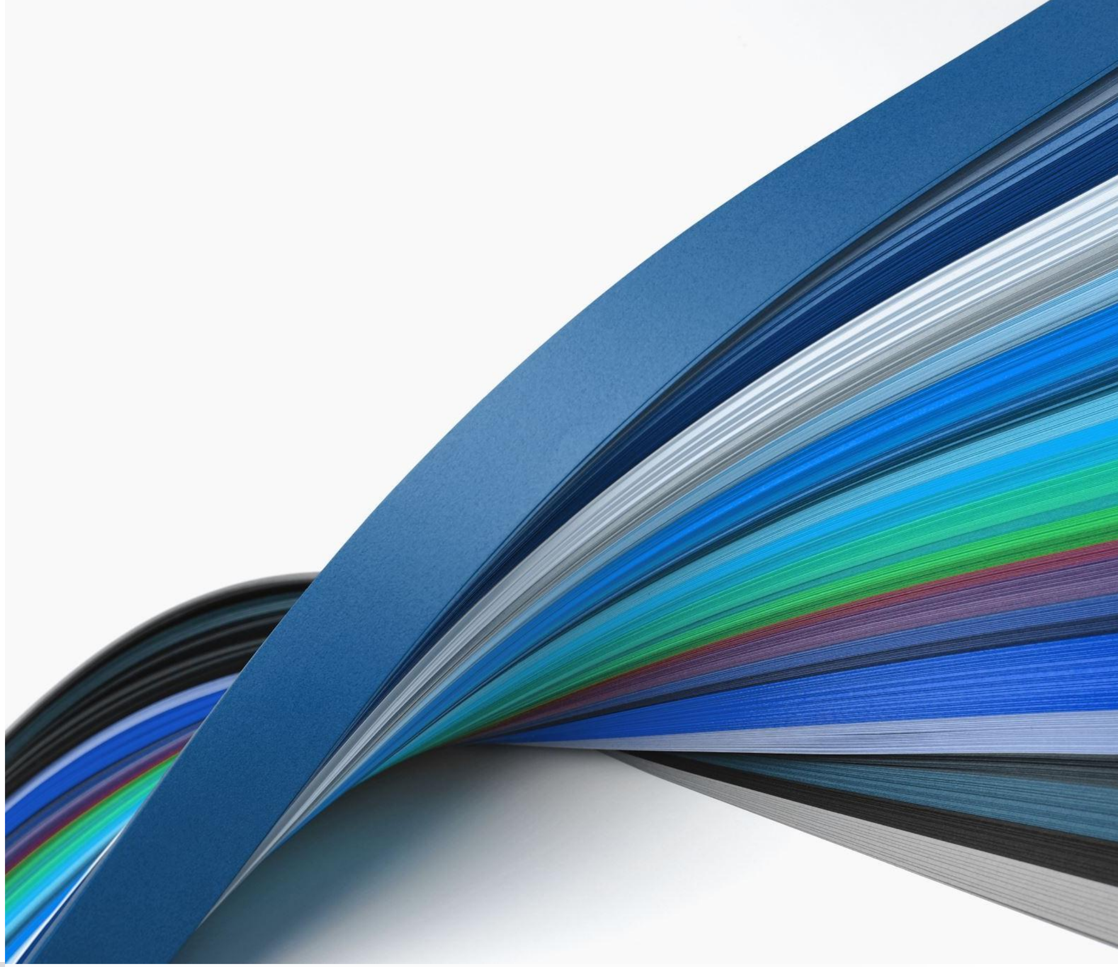
Chatbots and Dialog Systems



Speech Language Processing

Faculty of Computer Science Universitas Indonesia

Semester Gasal 2024/2025

Dr. Kurniawati Azizah, S.T., M.Phil.

References

- Dialog Systems Overview: Human conversations. Task-oriented dialog – Andrew Maas

Dialog system: Outline

- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
    - Finite State dialog architecture
    - Frame-based dialog architecture
    - The dialog-state architecture

Dialogue Systems and Chatbots

- Personal Assistants on phones or other devices
    - SIRI, Alexa, Cortana, Google Assistant
- Playing music, setting timers, reading recipes
- Booking reservations
- Answering questions
- Creative writing
- Editing or rewriting text
- Writing code

Two classes of systems

Chatbots

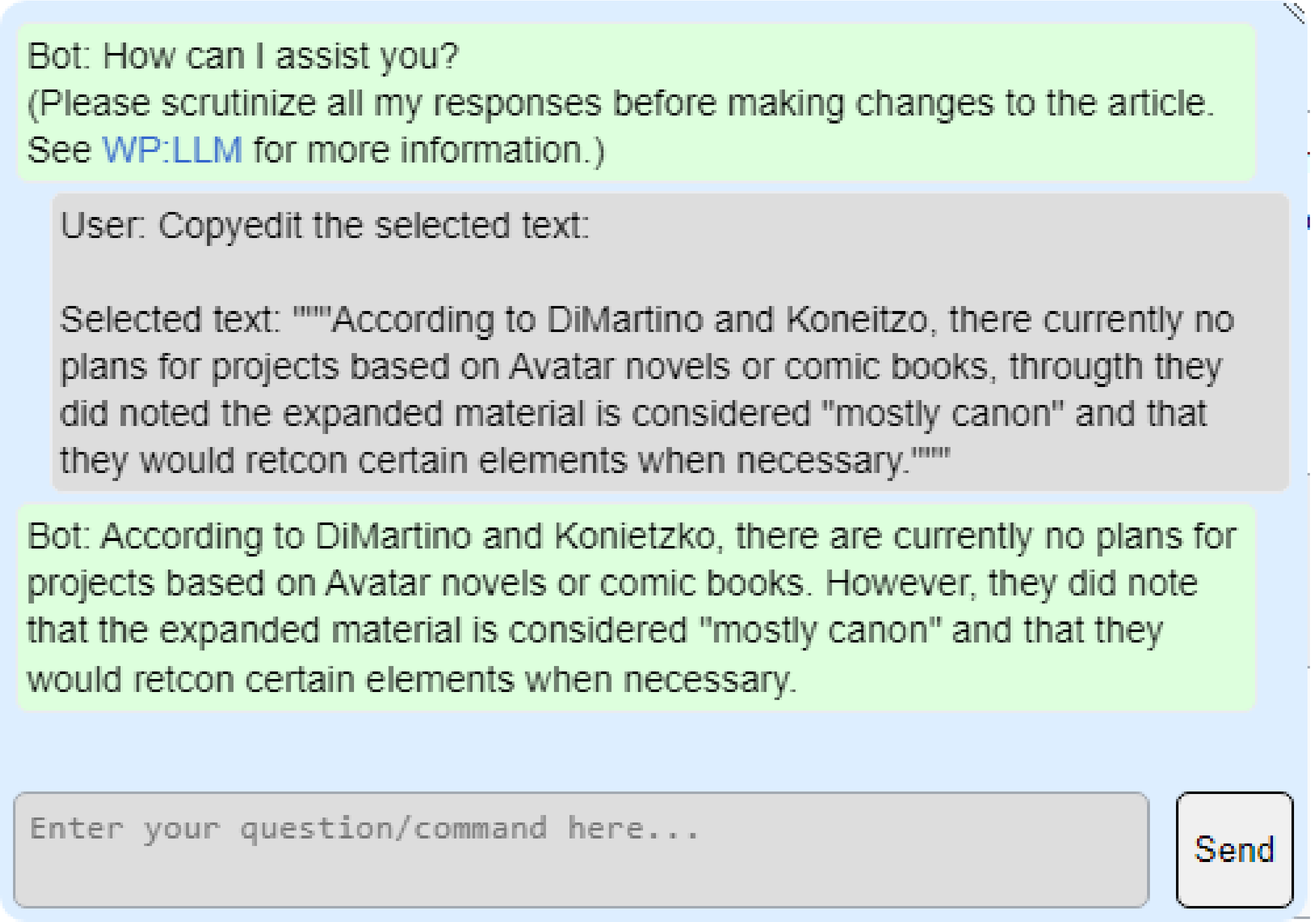
- Not goal/task-oriented. Just chat. No actions
- Open-ended, broad domain – chat about anything
- Classic metric: Turing test. Indistinguishable from human

Dialog systems

- Goal oriented. Actions could be API calls to web services
- Often domain/task-specific.
- Classic metric: Is the task completed properly?
- Human-like chat along with way is nice-to-have

Chatbots

Fun conversation, not tied to actions in the world or grounded in factual information in all cases



Bot: How can I assist you?
(Please scrutinize all my responses before making changes to the article. See WP:LLM for more information.)

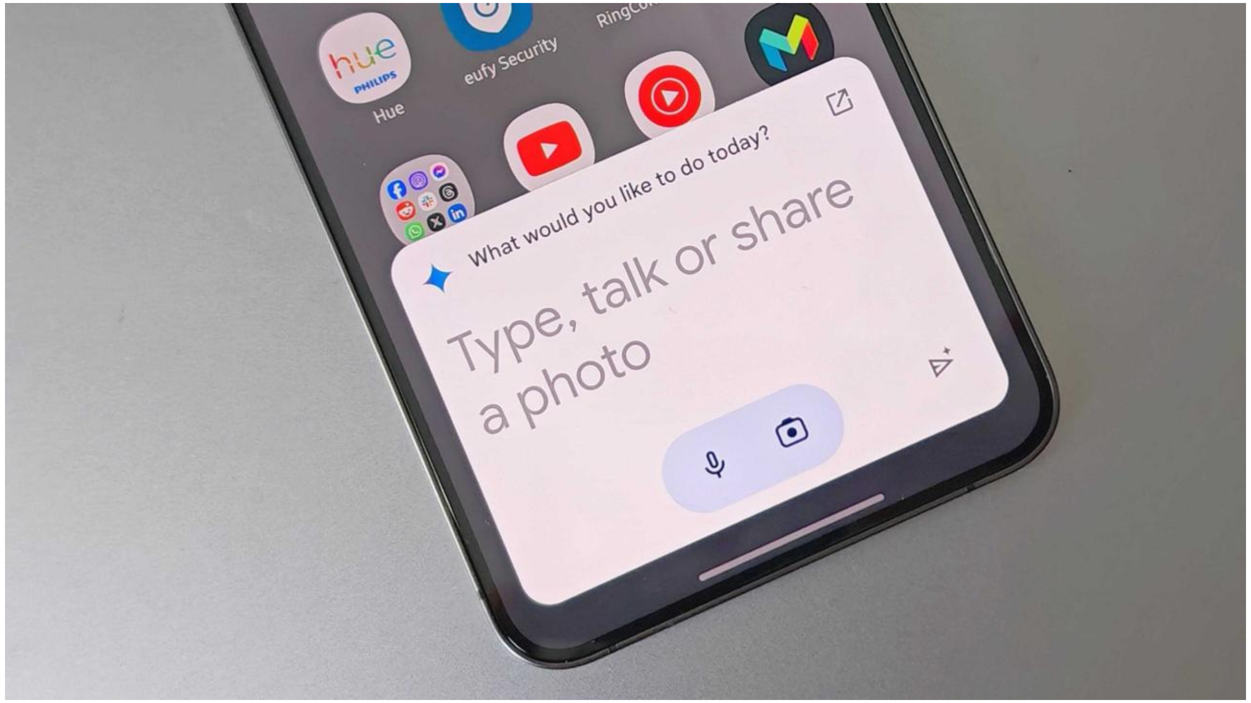
User: Copyedit the selected text:

Selected text: """According to DiMartino and Koneitzo, there currently no plans for projects based on Avatar novels or comic books, througth they did noted the expanded material is considered "mostly canon" and that they would retcon certain elements when necessary."""

Bot: According to DiMartino and Konietzko, there are currently no plans for projects based on Avatar novels or comic books. However, they did note that the expanded material is considered "mostly canon" and that they would retcon certain elements when necessary.

Conversational / dialog agents

- Applications:
  - Travel arrangements (Amtrak, United airlines)
  - Telephone call routing
  - Tutoring
  - Communicating with robots
  - Anything with limited screen/keyboard



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?

Dialogue agents based on large language models

- Like ChatGPT: based on large language models like GPT pretrained to predict words.
- These language models are fine-tuned to carry on conversation and follow instructions
- They can also retrieve text as part of answering questions or chatting
  - retrieval-augmented generation (RAG)

Dialog system: Outline

- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
    - Finite State dialog architecture
    - Frame-based dialog architecture
    - The dialog-state architecture

Task-oriented human conversation

- Turn-taking
- speech acts
- grounding

Human: . . . I need to travel in May.
Agent: And, what day in May did you want to travel?
Human: OK uh I need to be there for a meeting that's from the 12th to the 15th.
Agent: And you're flying into what city?
Human: Seattle
Agent: And what time would you like to leave Pittsburgh?
Human: Uh hmm I don't think there's many options for non-stop.
Agent: Right. There's three non-stops today.

Turn-taking

- Dialogue is characterized by turn-taking:

A: ""
B: ""
A: ""
B: ""

How do speakers know when to take the floor? Adjacency Pairs

Adjacency Pairs

- Current speaker selects next speaker
  - Question/answer
  - Greeting/greeting
  - Compliment/downplayer
  - Request/grant

- Silence inside the pair is meaningful

A: Is there something bothering you or not? (1.0)
A: Yes or no? (1.5)
A: Eh
B: No

Sacks et. al, 1974

Speech acts

Austin (1962): An utterance is a kind of action

- Clear case: performatives
  - I name this ship the Titanic
  - I second that motion
  - I bet you five dollars it will snow tomorrow
- Performative verbs
  - name, second
- Locutionary (what was said)
- Illocutionary (what was meant)

Five classes of speech acts

Searle (1975)

- Assertives:
  - Committing the speaker to something is being the case
  - Suggesting, putting forward, swearing, boasting, concluding
- Directives:
  - Attempts by speaker to get addressee to do something
  - Asking, ordering, requesting, inviting, advising, begging
- Commissives:
  - Committing speaker to future course of action
  - Promising, planning, vowing, betting, opposing

Five classes of speech acts

Searle (1975)

- Expressives:
  - Expressing psychological state of the speaker about a state of affairs
  - Thanking, apologizing, welcoming, deploring
- Declarations:
  - Changing the world via the utterance
  - I resign; You’re fired

Generalized idea – dialog acts

- Also called “conversational moves”
- An act with (internal) structure related specifically to its dialogue function
- Ontology varies by task, domain, etc. but core concepts like grounding often present

THANK	Thanks
GREET	Hello Dan
INTRODUCE	It’s me again
BYE	All right, bye
REQUEST-COMMENT	How does that look?
SUGGEST	June 13th through 17th
REJECT	No, Friday I’m booked all day
ACCEPT	Saturday sounds fine
DELIBERATE	Let me check my calendar here
CONFIRM	Okay, that would be wonderful
CLARIFY	Okay, do you mean Tuesday the 23rd

Grounding

- Need to achieve common ground
- Need to know whether an action succeeded or failed
- Hearer must ground or acknowledge speaker's utterance

How Do Speakers Ground?

- Continued attention: B continues attending to A
- Relevant next contribution: B starts in on next relevant contribution
- Acknowledgement: B nods or says continuer (uh-huh) or assessment (great!)
- Demonstration: B demonstrates understanding A by reformulating A's contribution, or by collaboratively completing A's utterance
- Display: B repeats verbatim all or part of A's presentation

Grounding Example

Display

Human: I need to travel In May
Agent: And, what day in May did you want to travel?

Acknowledgement

Human: I want to fly from Boston
Agent: mm-hmm
Human: to Baltimore Washington International

Acknowledgement + relevant next contribution

Human: I can leave on the 17th
Agent: And, what time would you like to leave?

Grounding negative response

Agent: Did you want to review some more of your personal profile?
Human: No
Agent: Okay, what’s next?

👍

Agent: Did you want to review some more of your personal profile?
Human: No
Agent: What’s next?

👎

Cohen et. al, 2004

Confirmation

Explicit

Human: I'd like to fly from Denver Colorado to New York City on September 21st in the morning on United Airlines
Agent: Let's see then. I have you going from Denver Colorado to New York on September 21st. Is that correct?
Human: Yes

Implicit

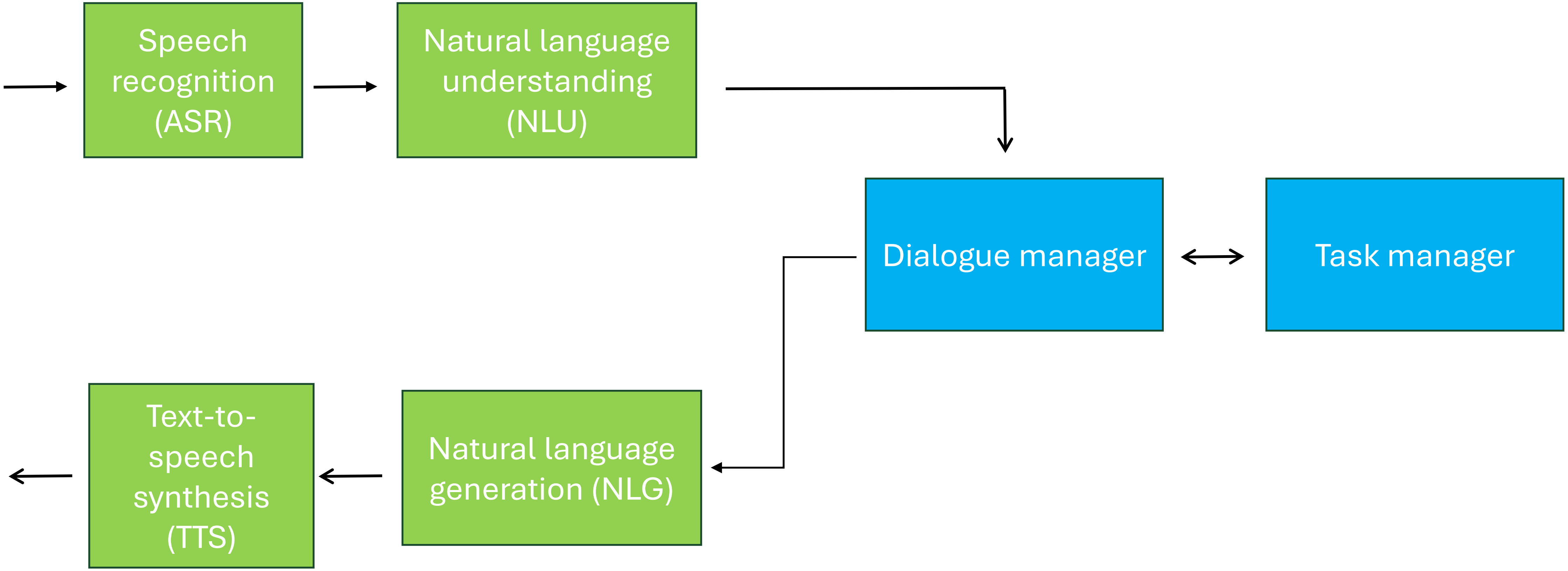
Human: I'd like to travel to Berlin
Agent: When do you want to travel to Berlin?

Human: Hi I'd like to fly to Seattle Tuesday morning
Agent: Traveling to Seattle on Tuesday, August eleventh in the morning. Your name?

Dialog system: Outline

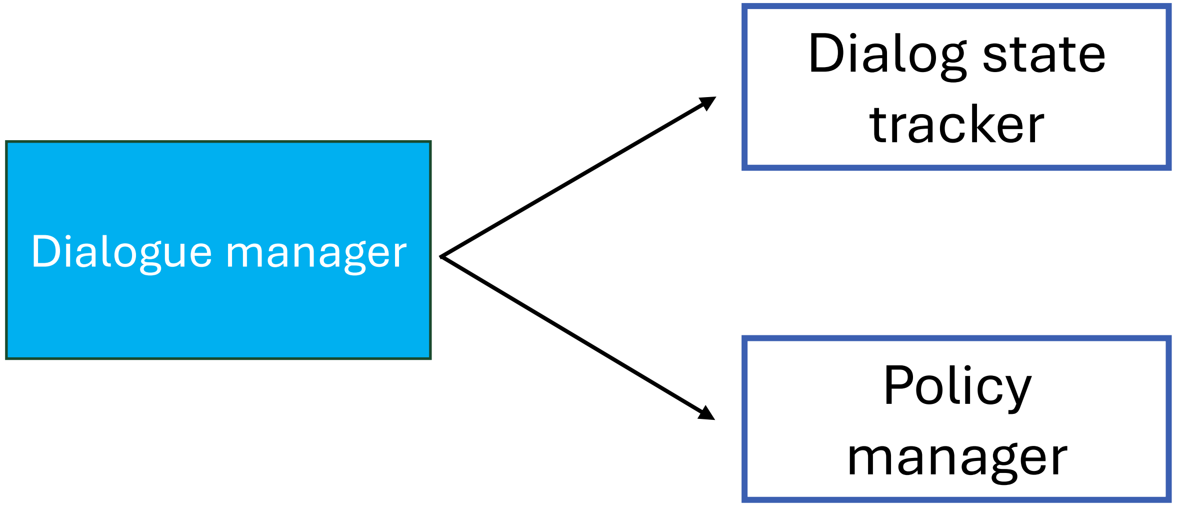
- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
    - Finite State dialog architecture
    - Frame-based dialog architecture
    - The dialog-state architecture

Spoken dialog agent conceptual architecture


Speech recognition (ASR)
Natural language understanding (NLU)
Dialogue manager
Task manager
Natural language generation (NLG)
Text-to-speech synthesis (TTS)

Dialogue + task management

- Control the architecture and structure of dialogue
- Decide what it knows, what to say next
- Takes input from ASR/NLU component and connect it with structured task information/state
- Maintain some sort of state of task and structured data (not just conversation history)
- Choose actions and send information to NLG to produce a response
- Take actions via task



Dialogue manager

Dialog state tracker

Policy manager

Dialogue initiative

- Who has control of conversation
- In normal human-human dialogue, initiative shifts back and forth between participants

Two types of the initiative :

1. **User initiative**
    - User asks a single question, system answers
    - Systems can not ask questions back

2. **System initiative**
    - System completely controls the conversation

Problems with system initiative

- Real dialogue involves give and take!
- In travel planning, users might want to say something that is not the direct answer to the question
- For example, answering more than one question in a sentence:

Human: I want a flight from Milwaukee to Orlando one way leaving after 5 p.m. on Wednesday

Examples of design considerations for dialogue acts

Confirmation

- Yes/no question
- ASR confidence is above threshold

Rejection

- “I’m sorry, I didn’t understand that”
- ASR confidence is low
- Best interpretation is semantically ill-formed

Implicit versus explicit confirmation

- Implicit: much more natural, quicker, simpler (if system guesses right)
- Explicit: easier for users to correct systems's mistakes (can just say "no")
- Early systems: all-implicit or all-explicit
- Modern systems: adaptive
- How to decide when to be explicit?
  - ASR system can give confidence metric on its transcription of the speech
  - If high confidence, use implicit confirmation
  - If low confidence, use explicit confirmation

Four-tiered level of confidence:

- Below confidence threshold, reject
- Above threshold, explicit confirmation
- If even higher, implicit confirmation
- Even higher, no confirmation

Conversational agent problem space

- Time to response (Synchronous?)
- Task complexity
  - What time is it?
  - Book me a flight and hotel for vacation in Greece
- Interaction complexity / number of turns
  - Single command/response
  - “I want new shoes” What kind? What color? What size?
- Initiative
  - User, System, Mixed
  - Interaction modality
- Purely spoken, Purely text, Mixing speech/text/media

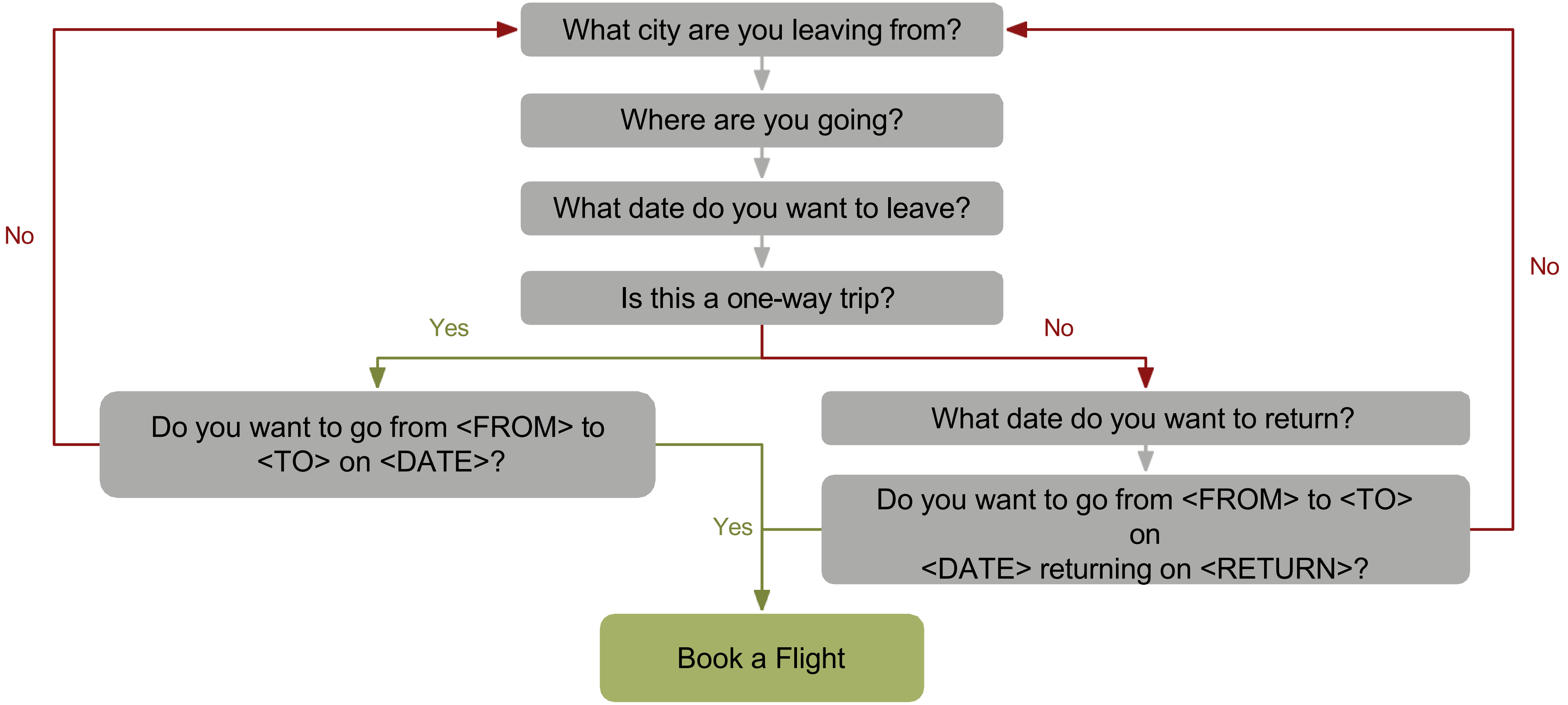
Dialog system: Outline

- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
    - Finite State dialog architecture
    - Frame-based dialog architecture
    - The dialog-state architecture

Possible Architectures for Dialog Management

- Finite State
- Frame-based
    - Alexa skills kit uses a version of this
- Information State (Markov Decision Process)
- Distributional / Neural Network

Finite State Dialog Manager



What city are you leaving from?

Where are you going?

What date do you want to leave?

Is this a one-way trip?

Yes

No

Do you want to go from <FROM> to <TO> on <DATE>?

What date do you want to return?

Do you want to go from <FROM> to <TO> on <DATE> returning on <RETURN>?

No

No

Yes

Book a Flight

Finite-State Dialog Managers

- System completely controls the conversation with the user.
- It asks the user a series of questions
- Ignoring (or misinterpreting) anything the user says
  that is not a direct answer to the system's questions
- Quick solution for simple tasks, scales poorly to complex/large tasks
- Consider a trivial airline travel system:
  - Ask the user for a departure city
  - Ask for a destination city
  - Ask for a time
  - Ask whether the trip is round-trip or not

Dialog system: Outline

- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
  - Finite State dialog architecture
  - Frame-based dialog architecture
  - The dialog-state architecture

Frame-based dialogue agents

Sometimes called "**task-based dialogue** agents"

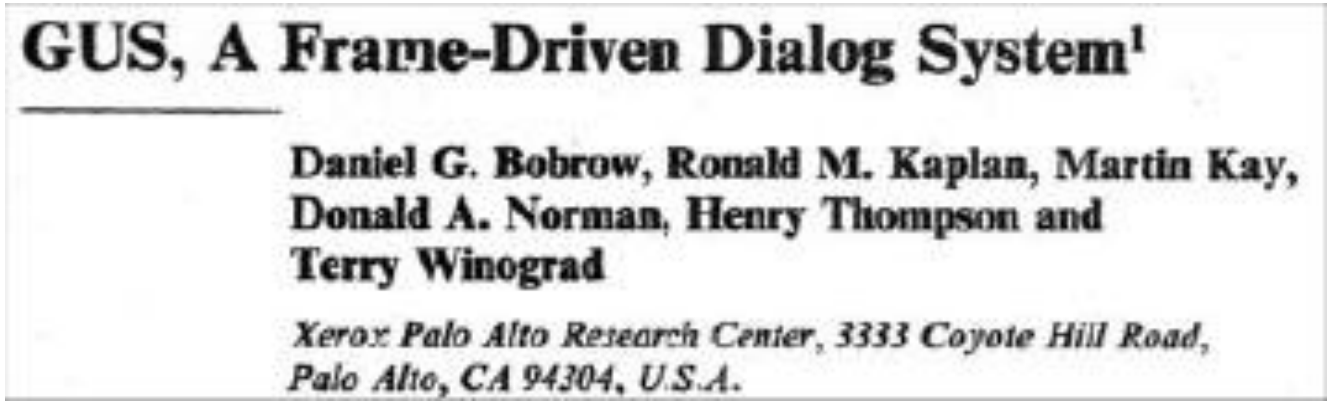
- Systems that have the goal of helping a user solve a task like making a travel reservation or buying a product

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**)

Frame-based dialogue agents

- 1977



GUS, A Frame-Driven Dialog System[1]

Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay,
Donald A. Norman, Henry Thompson and
Terry Winograd

Xerox Palo Alto Research Center, 3333 Coyote Hill Road,
Palo Alto, CA 94304, U.S.A.

- Still the industrial state of the art
- Amazon Alexa roughly follows GUS architecture

The frame

- A set of slots, to be filled with information of a given slot 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?

System Design Considerations

- One or more **frames**
  - Each a collection of **slots**
  - Each slot having a **value**
- **Domain ontology** is full collection of frames for a (sub)system
- A frame defines the task data structure

Two basic architectures

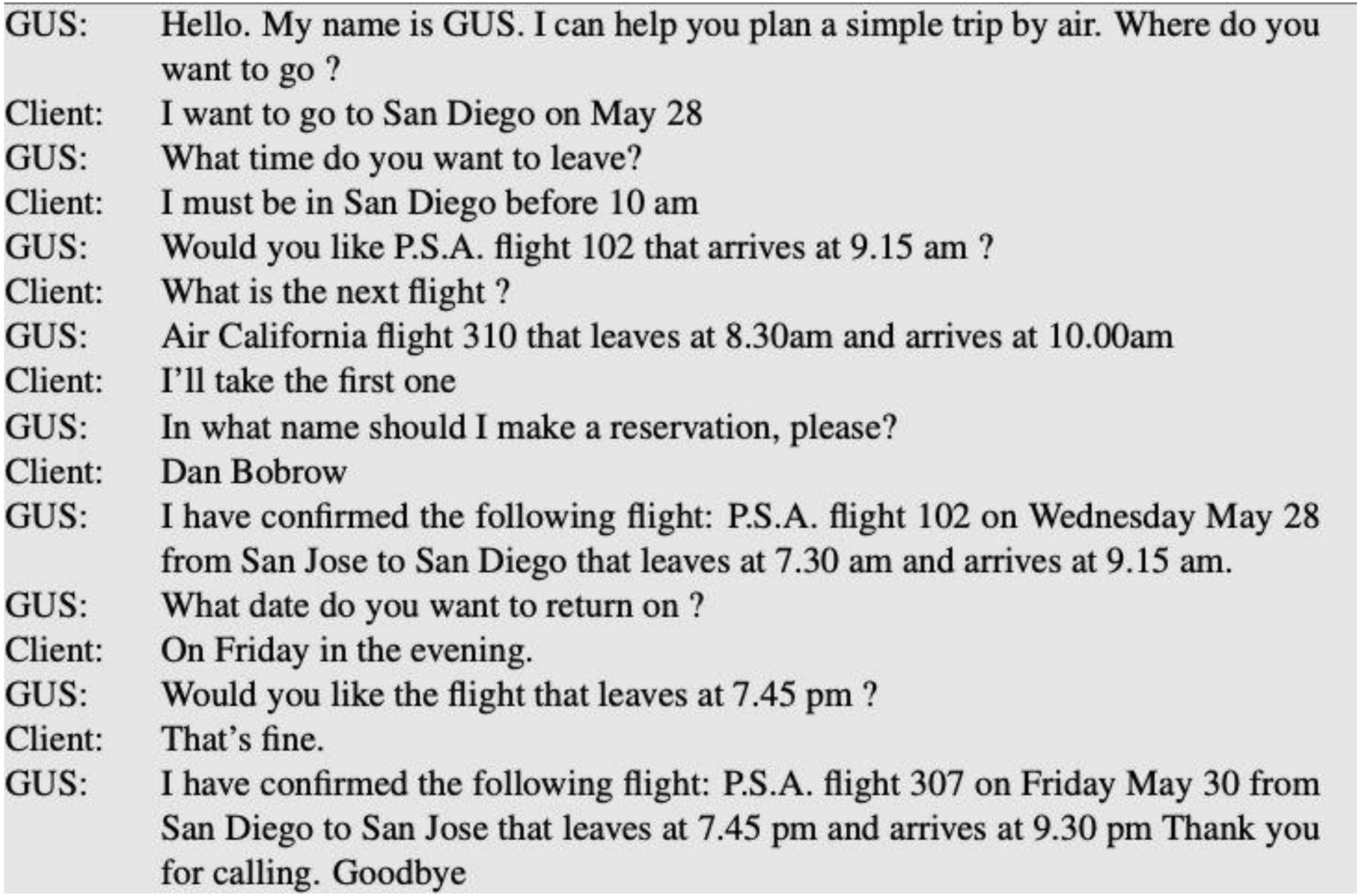
The **GUS** architecture

- 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

GUS - the State of the Art in 1677



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

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

1. 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:

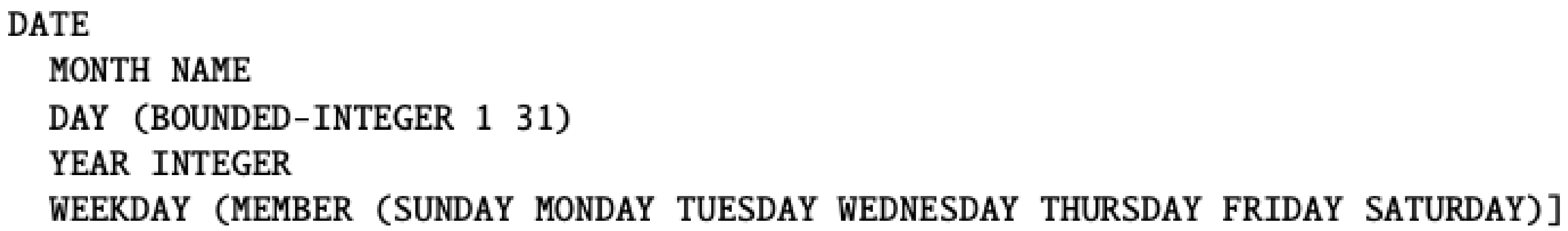
- Car or hotel reservations
- General route information
  - *Which airlines fly from Boston to San Francisco?,*
- Information about airfare practices
  - *Do I have to stay a specific number of days to get a decent airfare?*).

Frame detection:

- System must detect which slot of which frame user is filling
- And switch dialogue control to that frame.

Slot Types Can Be Complex, Hierarchical

- The type
  DATE



```
DATE
  MONTH NAME
  DAY (BOUNDED-INTEGER 1 31)
  YEAR INTEGER
  WEEKDAY (MEMBER (SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY)]
```

Frames and mixed initiative

- System asks questions of user, filling any slots that user specifies
  - When frame is filled, do database query
- If user answers 3 questions at once, system can fill 3 slots and not ask these questions again!
- Frame structure guides dialog
- Simplest kind of mixed initiative: use the structure of the frame to guide dialogue

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?

Natural language understanding for filling dialog slots

- Domain classification
    - Asking weather? Booking a flight? Programming alarm clock?
- Intent Determination
    - Find a Movie, Show Flight, Remove Calendar Appointment
- Slot Filling
    - Extract the actual slots and fillers

Natural language understanding for filling slots

Human: Show me morning flights from Boston to San Francisco on Tuesday

Dialogue manager:

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

Natural language understanding for filling slots

Human: Turn on my alarm for 6am on May 28

Dialogue manager:

```
DOMAIN:    ALARM-CLOCK
INTENT:    SET-ALARM
TIME:       2024-05-28 0600
```

GUS: Rule-based slot-filling

- Write regular expressions or grammar rules

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

- Do text normalization
- Time consuming and brittle NLU capabilities
- With modern NLP tools/features, only use rules alone in special cases
- Simple rules + LLM few-shot recognizers might be just as easy and more robust

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

Dialog system: Outline

- Overview
- Human conversation
- Dialog system conceptual architecture
- Possible dialog architectures
  - Finite State dialog architecture
  - Frame-based dialog architecture
  - The dialog-state architecture

Dialogue-State or Belief-State Architecture

A more sophisticated version of the frame-based architecture

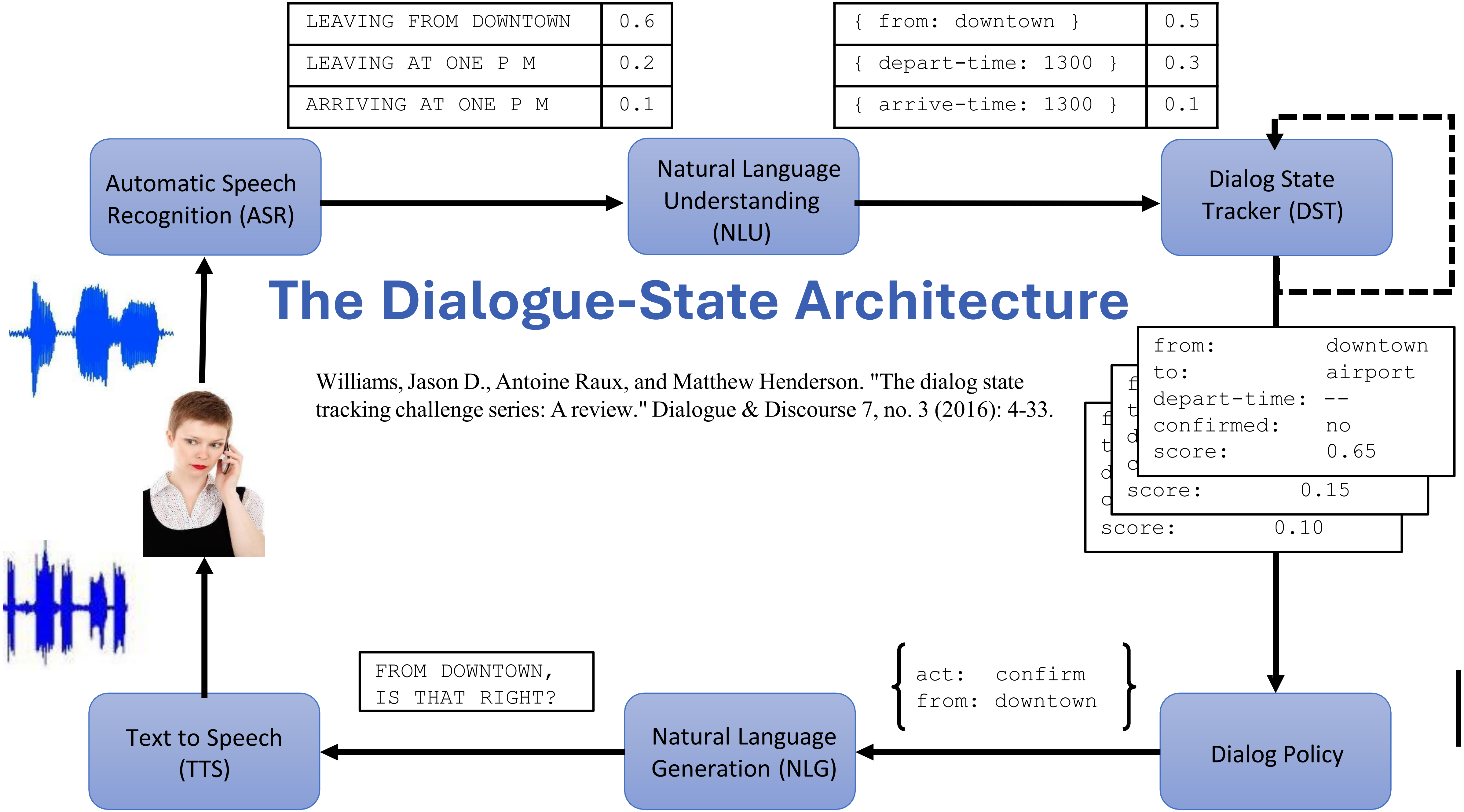
- Has dialogue acts, more ML, better generation

  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

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


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

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

```
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?
```

Components in a dialogue-state architecture

NLU: extracts slot fillers from the user's utterance using machine learning

Dialogue state tracker (DST) : 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

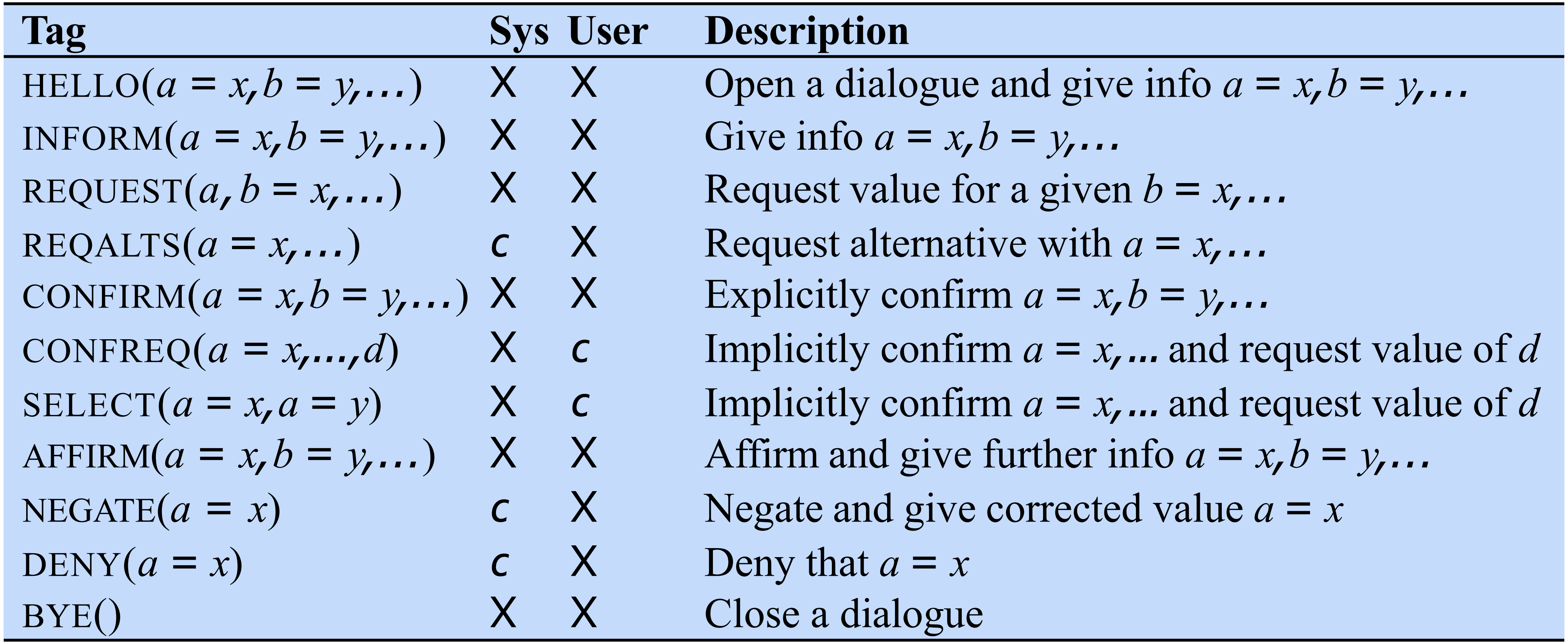
- GUS policy: ask questions until the frame was full then report back
- More sophisticated: know when to answer questions, when to ask a clarification question, etc.

NLG: produce more natural, less templated utterances

Dialogue Acts

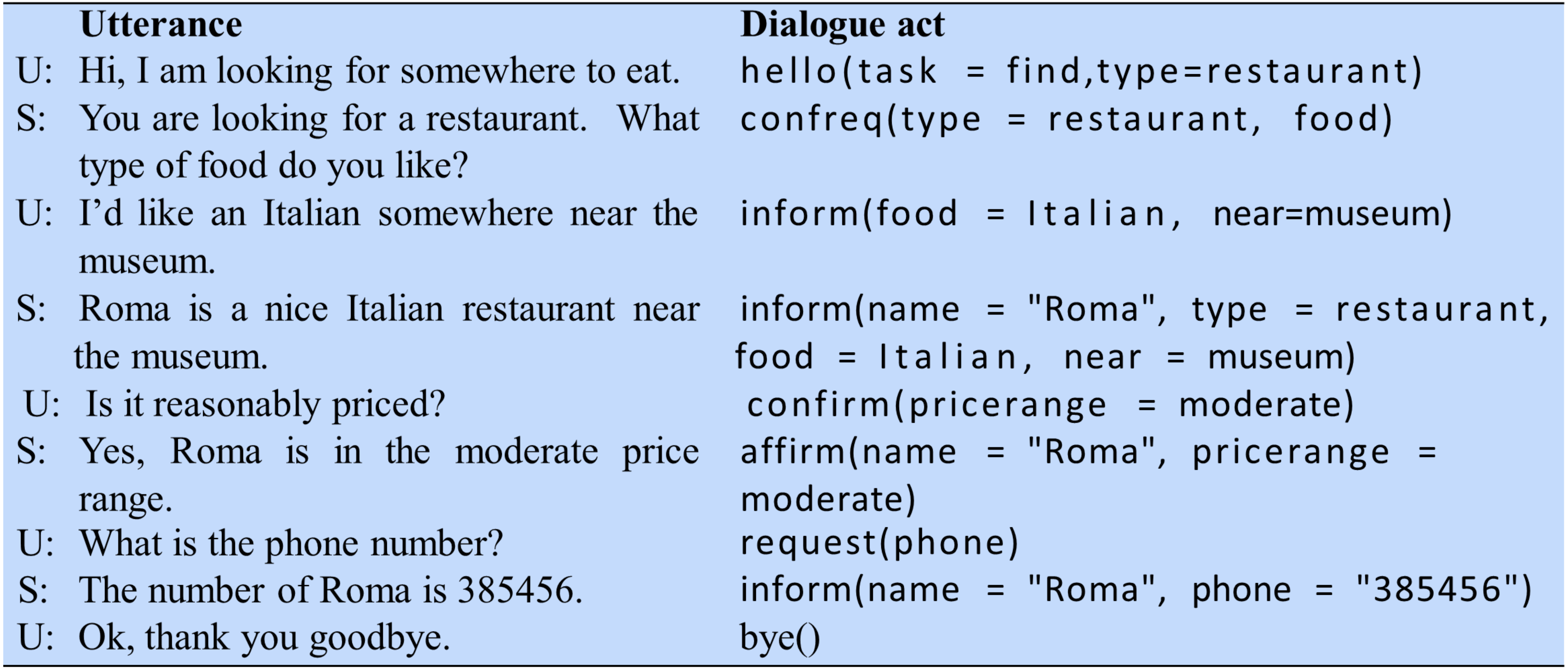
Combine the ideas of **speech acts** and **grounding** into a single representation

Young et al., 2010:



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

Dialogue Acts



	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()

Machine Learning for Slot-Filling

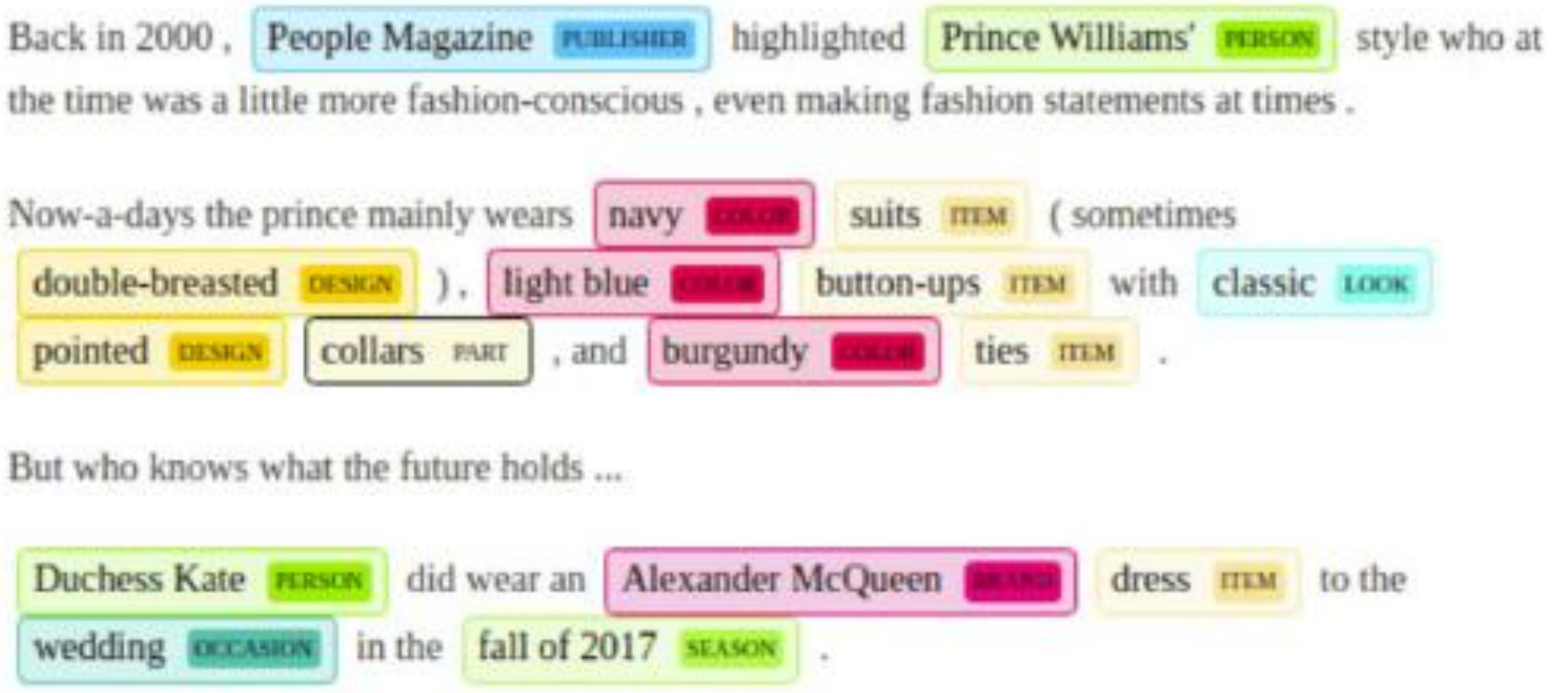
`I want to fly to San Francisco on Monday afternoon please`

- Use 1-of-N classifier for Domain/Intent. Use sequence model to tag words/phrases with slot name
- Input: features like word N-grams
- Output:

```
DOMAIN:       AIRLINE
INTENT:       SHOW-FLIGHT
DESTINATION: “San Francisco”
DEPART-DATE: “Monday”
```

Text natural language processing for filling slots

Conditional Random Field (CRF) with word vector features, or neural classifiers both work well



Back in 2000 , People Magazine highlighted Prince Williams' style who at the time was a little more fashion-conscious , even making fashion statements at times .

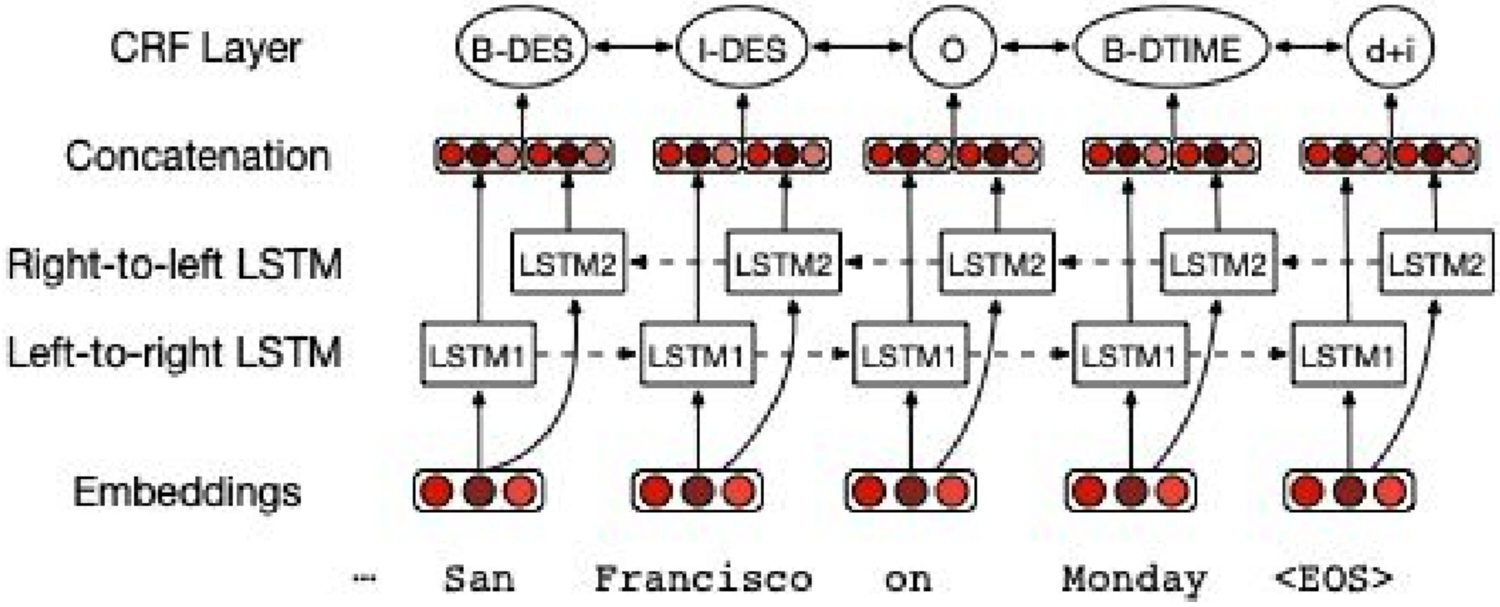
Now-a-days the prince mainly wears navy suits (sometimes double-breasted) , light blue button-ups with classic pointed collars , and burgundy ties .

But who knows what the future holds ...

Duchess Kate did wear an Alexander McQueen dress to the wedding in the fall of 2017 .

Sequence Models for Slot-Filling: BIO Tagging

- BIO Tagging is done by a sequence model



- Extracted strings can then be normalized (San Fran->SFO)

Intent detection and slot filling using BIO labelling format.

The "B-" prefix indicates that the word is at the beginning of a slot.

The "I-" prefix indicates that the word is on the inside or at the end of a slot (occurs when slots contain at least 2 words).

The "O" label is assigned to words not belonging to any slot.

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

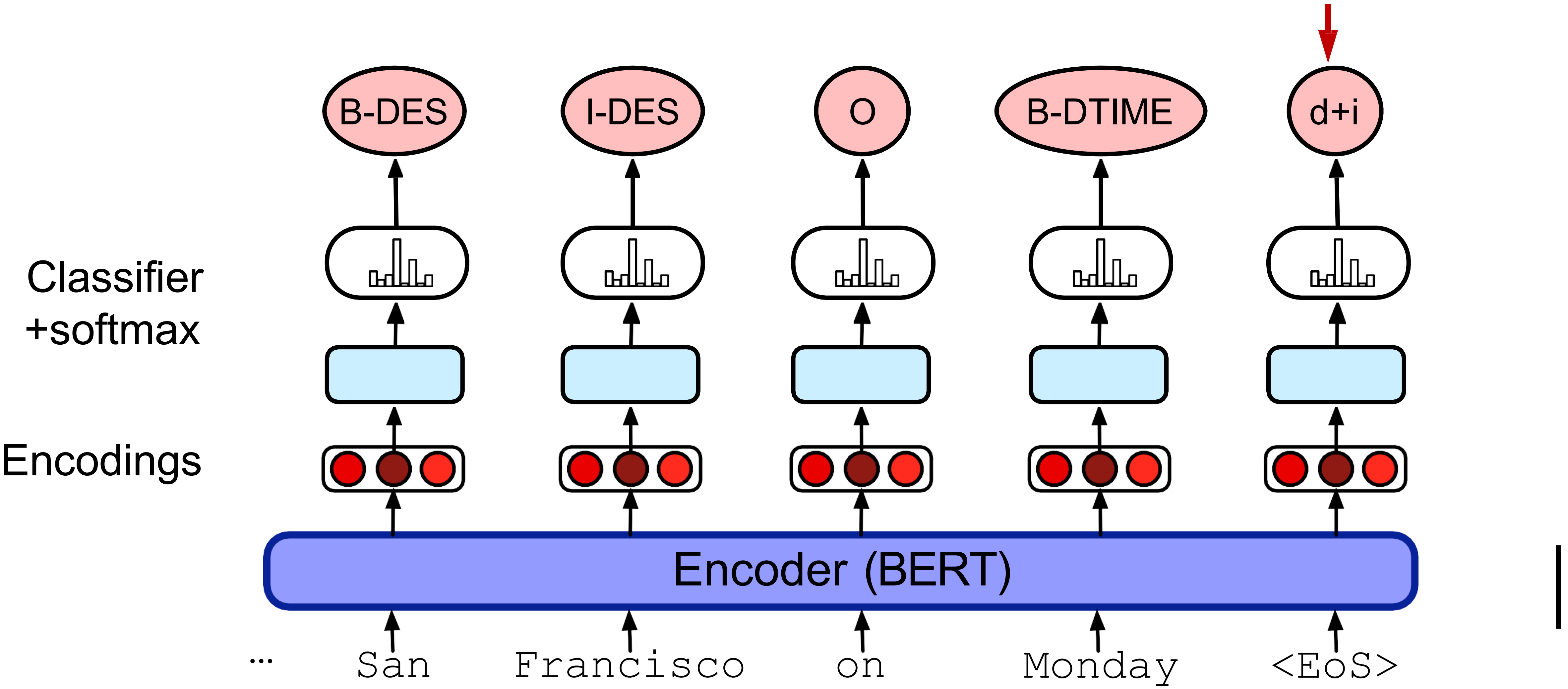
O	O	O	O	O	B-DES	I - DES	O	B-DEPTIME	I - DEPTIME	O
I	want	to	fly	to	San	Francisco	on	Monday	afternoon	please

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., "AIRLINE_TRAVEL + SEARCH_FLIGHT"

generate the label



Once we have the BIO tag of the sentence

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 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)

Dialogue state tracking (DST)

I'd like Cantonese food near the Mission district.

→

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

Dialogue act interpretation algorithm:

- 1-of-N supervised classification to choose `inform`
- Based on encodings of current sentence + prior dialogue acts Simple dialogue state tracker:
- Run a slot-filler after each sentence

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

Example from Mrkšić, N., O Séaghdha, D., Wen, T.-H., Thomson, B., and Young, S. (2017). Neural belief tracker: Data-driven dialogue state tracking. *ACL*.

An special case of dialogue act detection: Detecting Correction Acts

If system misrecognizes an utterance

User might make a **correction**

- Repeat themselves
- Rephrasing
- Saying “no” to a confirmation question

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:
  - Shriberg, E., Wade, E., Price, P., 1992. Human-machine problem solving using spoken language systems (SLS): Factors affect-ng performance and user satisfaction. DARPA Speech and Natural Language Workshop.
- "I said BAL-TI-MORE, not Boston"

Features for detecting corrections in spoken dialogue

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 Policy

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

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

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

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

Policy example: Confirmation and Rejection

Dialogue systems make errors

So they to make sure they have understood user Two important mechanisms:

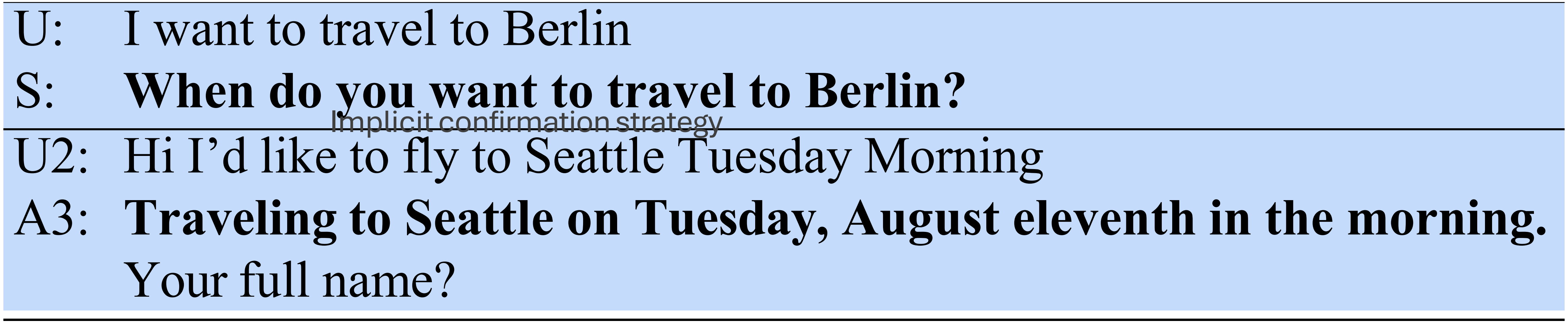
- **confirming** understandings with the user
- **rejecting** utterances that the system is likely to have misunderstood.

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?**

Implicit confirmation strategy

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).

Rejection

I’m sorry, I didn’t understand that.

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:

$< a$	low confidence	reject
$\geq a$	above the threshold	confirm explicitly
$\geq b$	high confidence	confirm implictly
$\geq g$	very high confidence	don’t confirm at all

Natural Language Generation (NLG)

NLG in information-state architecture modeled in two stages:

- **content planning** (what to say)
  - ask a question, present an answer, etc
  - Often merged with dialogue manager
- **sentence realization** (how to say it).
  - chooses syntax and words
- **In practice**: template-based with most words prespecified:

What time do you want to leave CITY-ORIG?

Will you return to CITY-ORIG from CITY-DEST?

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)

2 samples of 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.

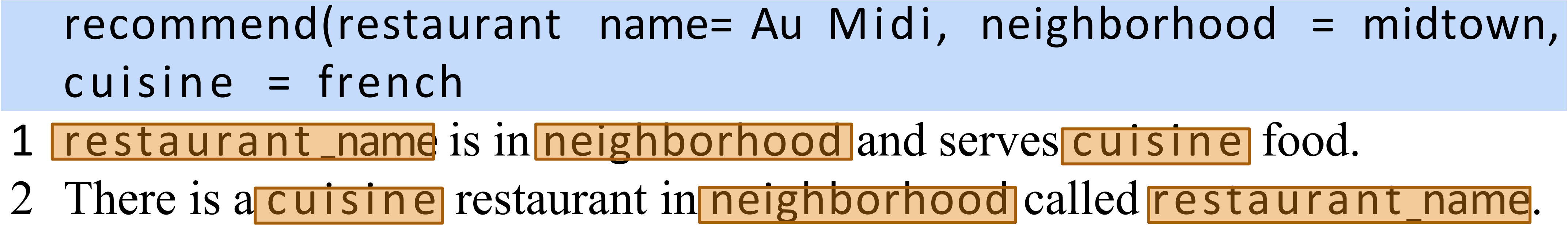
Sentence Realization

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:

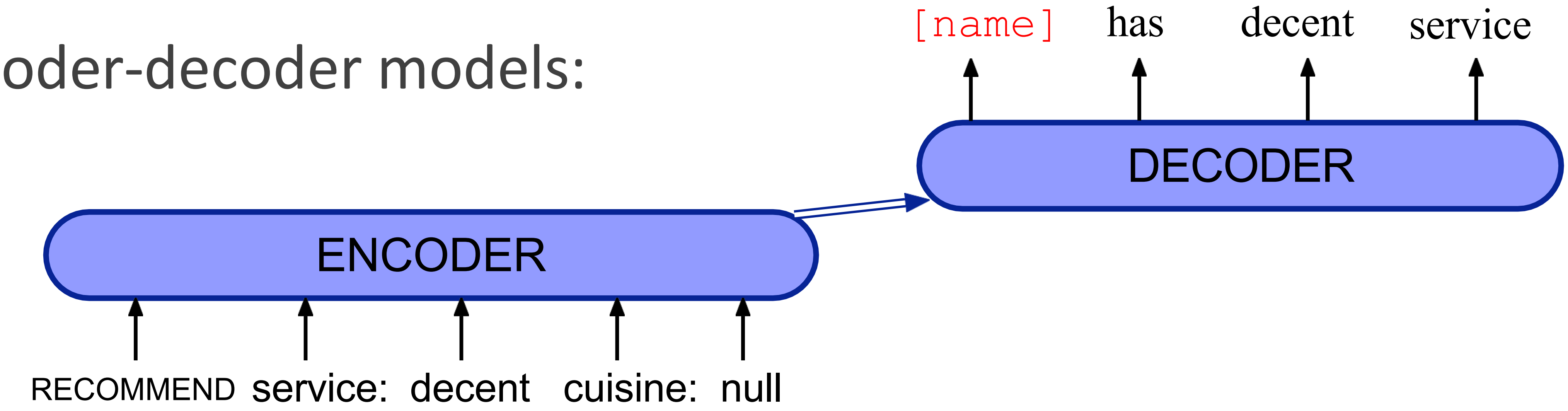


recommend(restaurant name= Au Midi, neighborhood = midtown, cuisine = french

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`

More Sophisticated NLG

- Dialogue manager builds representation of meaning of utterance to be expressed
- Passes this to a “generator”. Old style was templates, modern systems use LLMs
- LLM-based NLG constrained to convey dialog representations can improve user satisfaction

- Critical aspect: Ensure correctness of what we convey to the user!

Deep Learning NLG Conditioned on Dialog Semantics

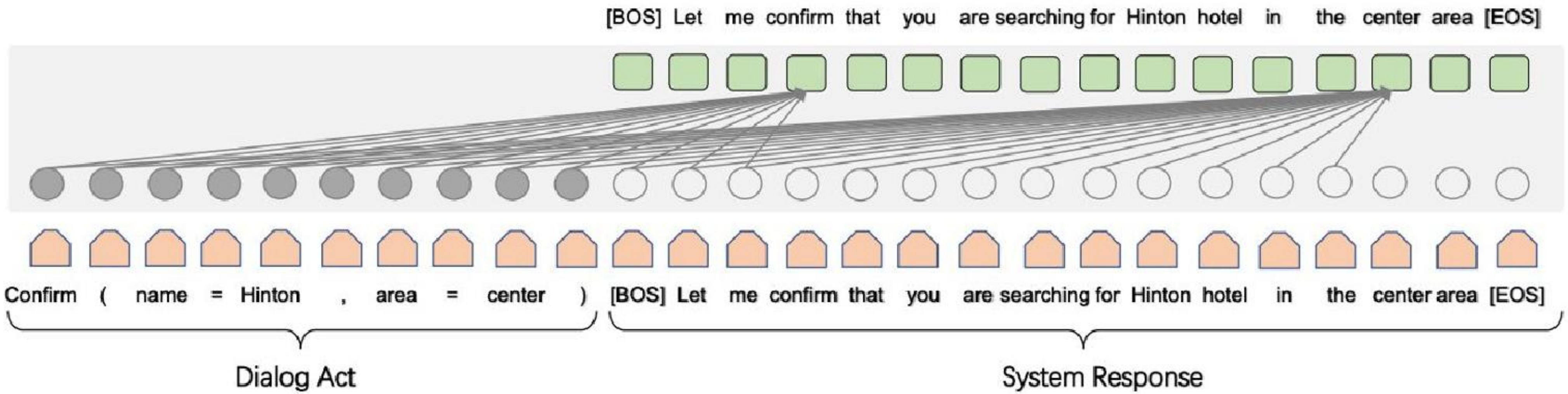

[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Confirm (name = Hinton , area = center)
[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Dialog Act
System Response

Figure: Illustration of SC-GPT. In this example, SC-GPT generates a new word token (e.g. “confirm” or “center”) by attending the entire dialogue act and word tokens on the left within the response. (Peng et al, 2020)

Semantically Conditioned GPT for Dialog NLG

$$\mathcal{A} = [\underbrace{\ \mathbf{I}\ }_{\text{Intent}}, \underbrace{(s_1, v_1), \cdots, (s_P, v_P)}_{\text{Slot-value pairs}}]$$

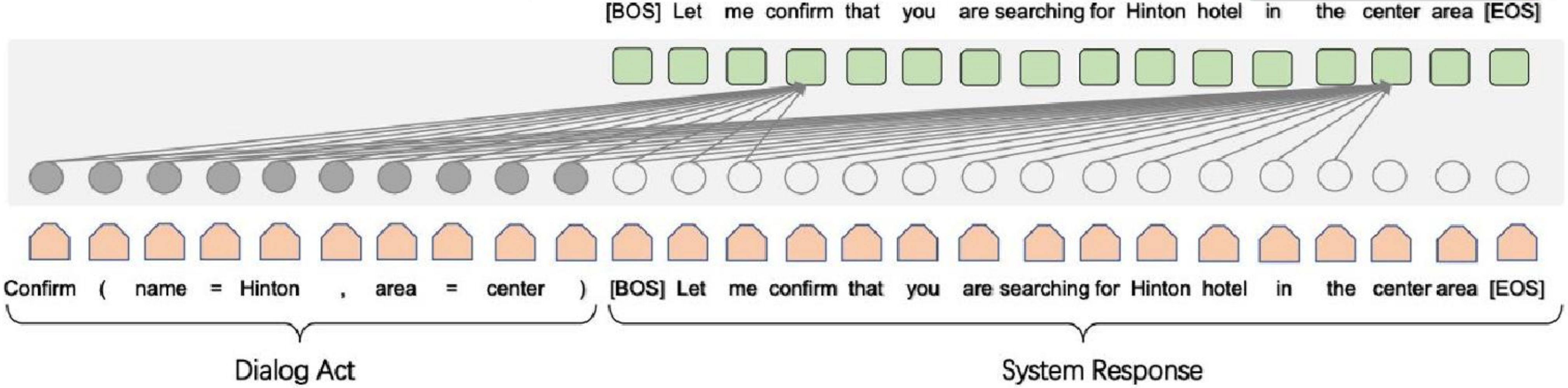
$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathcal{A}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|x_{<t}, \mathcal{A})$$

Dialog Act

Intent: *Confirm*
Slot-value pairs:
[*name = Hilton*], [*area = center*]

Response

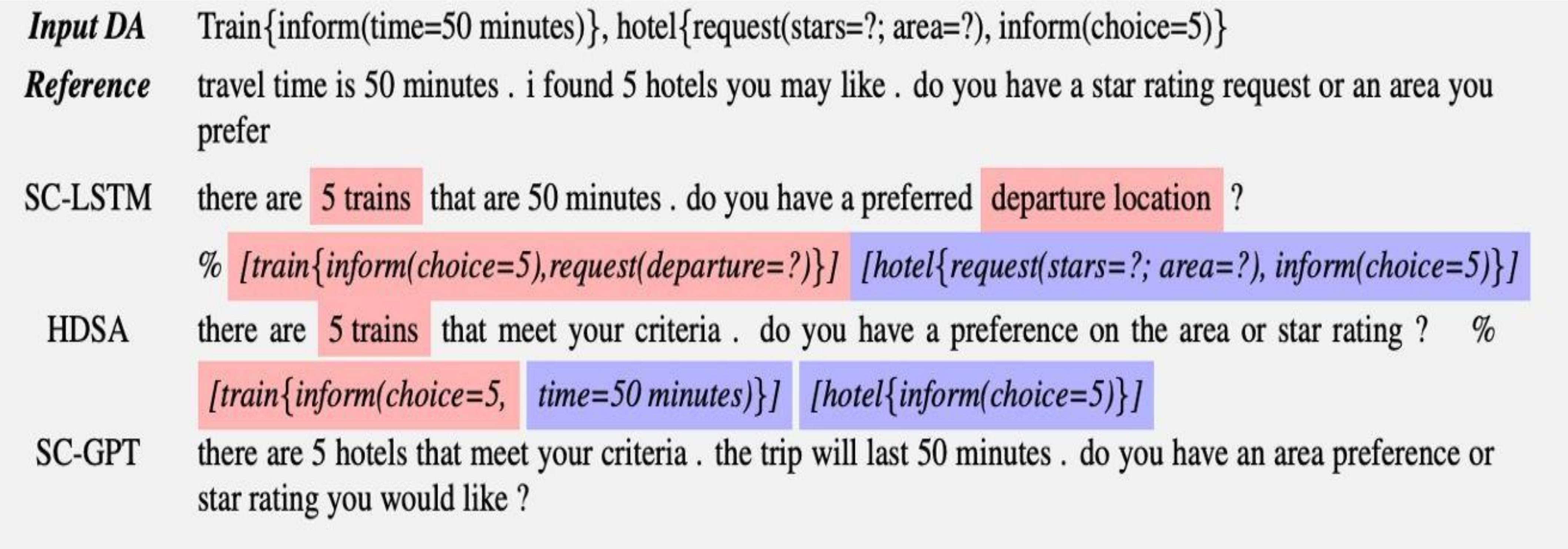
Let me confirm that
you are searching for Hinton hotel
in the center area



[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Confirm (name = Hinton , area = center) [BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Dialog Act
System Response

Figure: Illustration of SC-GPT. In this example, SC-GPT generates a new word token (e.g. “confirm” or “center”) by attending the entire dialogue act and word tokens on the left within the response. (Peng et al, 2020)

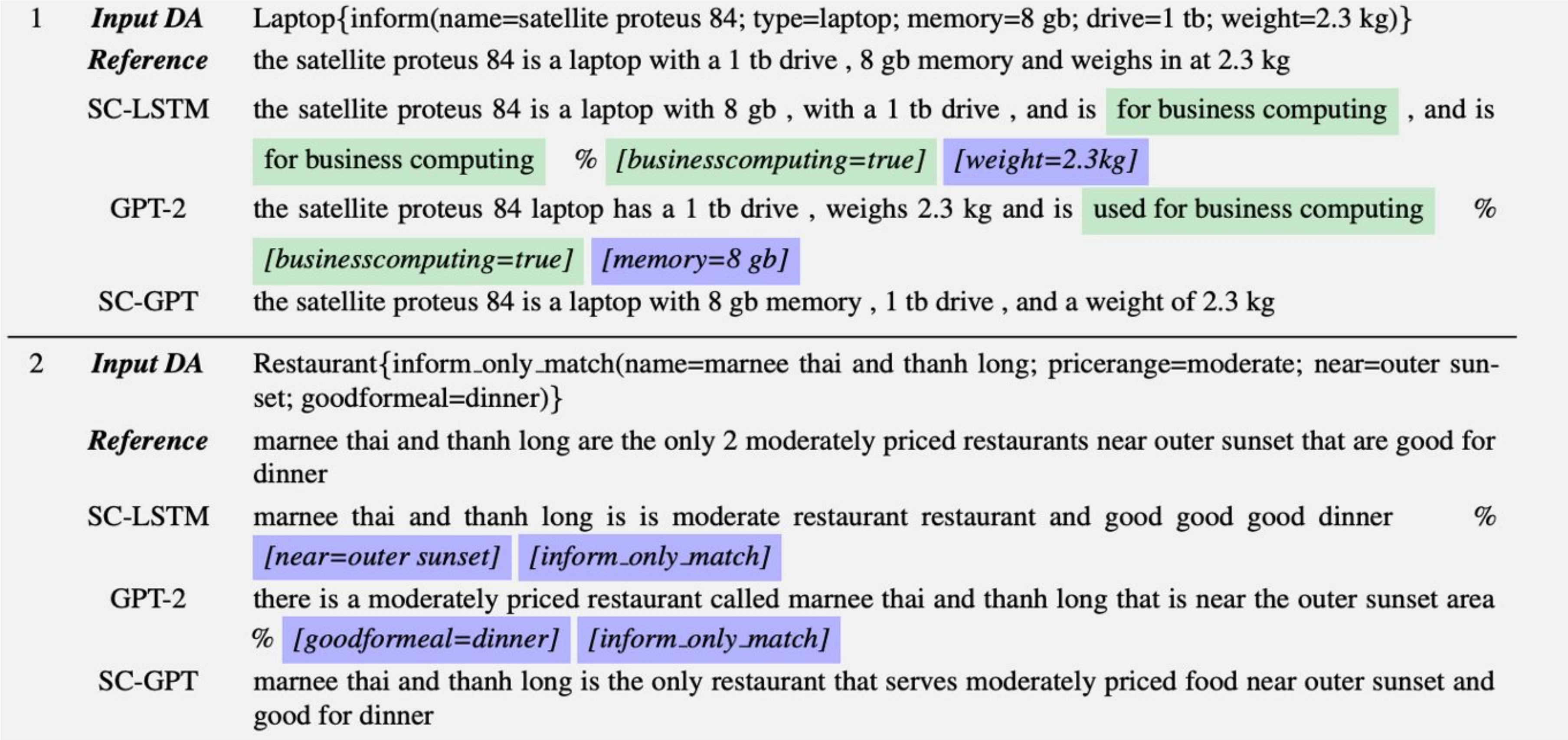
Deep Learning NLG Conditioned on Dialog Semantics



Input DA	Train{inform(time=50 minutes)}, hotel{request(stars=?; area=?), inform(choice=5)}
Reference	travel time is 50 minutes . i found 5 hotels you may like . do you have a star rating request or an area you prefer
SC-LSTM	there are 5 trains that are 50 minutes . do you have a preferred departure location ? *% [train{inform(choice=5),request(departure=?)}] [hotel{request(stars=?; area=?), inform(choice=5)}]*
HDSA	there are 5 trains that meet your criteria . do you have a preference on the area or star rating ? *%* *[train{inform(choice=5, time=50 minutes)}] [hotel{inform(choice=5)}]*
SC-GPT	there are 5 hotels that meet your criteria . the trip will last 50 minutes . do you have an area preference or star rating you would like ?

Examples of generated utterances from different models along with its corresponding dialogue acts (DAs) and references. Sampled from MultiWOZ. Each generated utterance is followed by a brief description explaining the errors (starting with %). Better viewed in color wrong , redundant , missing information. (Peng *et al*, 2020)

Deep Learning NLG Conditioned on Dialog Semantics



1	***Input DA***	Laptop{inform(name=satellite proteus 84; type=laptop; memory=8 gb; drive=1 tb; weight=2.3 kg)}
	Reference	the satellite proteus 84 is a laptop with a 1 tb drive , 8 gb memory and weighs in at 2.3 kg
	SC-LSTM	the satellite proteus 84 is a laptop with 8 gb , with a 1 tb drive , and is for business computing , and is for business computing % *[businesscomputing=true]* *[weight=2.3kg]*
	GPT-2	the satellite proteus 84 laptop has a 1 tb drive , weighs 2.3 kg and is used for business computing % *[businesscomputing=true]* *[memory=8 gb]*
	SC-GPT	the satellite proteus 84 is a laptop with 8 gb memory , 1 tb drive , and a weight of 2.3 kg
2	***Input DA***	Restaurant{inform_only_match(name=marnee thai and thanh long; pricerange=moderate; near=outer sunset; goodformeal=dinner)}
	Reference	marnee thai and thanh long are the only 2 moderately priced restaurants near outer sunset that are good for dinner
	SC-LSTM	marnee thai and thanh long is is moderate restaurant restaurant and good good good dinner % *[near=outer sunset]* *[inform_only_match]*
	GPT-2	there is a moderately priced restaurant called marnee thai and thanh long that is near the outer sunset area % *[goodformeal=dinner]* *[inform_only_match]*
	SC-GPT	marnee thai and thanh long is the only restaurant that serves moderately priced food near outer sunset and good for dinner

Examples of generated utterances from different models along with its corresponding dialogue acts (DAs) and references. The examples are sampled from FewShotWOZ. Each generated utterance is followed by a brief description explaining the errors (starting with %). Better viewed in color wrong , redundant , missing information. (Peng *et al*, 2020)