Dialogue Systems – Lecture 2

Bill Byrne

Lent 2022

Neural Machine Translation and Dialogue Systems – MLMI8

MPhil in Machine Learning and Machine Intelligence

UNIVERSITY OF CAMBRIDGE

Outline for Today’s Lecture

- Introduction to Dialogue Systems and Dialogues
    - types of dialogues and dialogue systems, their architecture
    - properties of dialogues, dialogue acts, hidden information states
- Task Oriented Dialogue System components and subtasks:
    - Natural Language Understanding
    - Dialogue State Tracking
    - Dialogue Policies
    - Natural Language Generation
- Sequence to Sequence models and Task Oriented Dialogue

General references for today’s and subsequent lectures

- Gao, Galley, Li, *Neural Approaches to Conversational AI Question Answering, Task-Oriented Dialogues and Social Chatbots,* 2019 https://arxiv.org/abs/1809.08267v3
- Jurafsky, Martin, *Speech and Language Processing*, 2019. Chatbots and Dialogue Systems, Chapter 26. https://web.stanford.edu/~jurafsky/slp3/24.pdf *
- Fung, Chen, Lin, Madotto, *Deeper Conversational AI.* NeurIPS 2020 Tutorial.

* new version

What is a dialogue system?

- A dialogue system is a computer system that enables human computer interaction where the primary input is spoken or written natural language.
- Speech needn’t be the only input:
  - We can interact with machines also by typing, touch, gesture, facial expressions, sensors, …
  - these are **multi-modal** dialogue systems.

Some systems capable of holding a dialogue:

alexa

Hey Cortana

Google Voice

Hey Siri

In this course, we’ll pretend we have perfect speech recognition – no errors or speech effects

What constitutes a dialogue system?

A dialogue system acts as an **agent** capable of conversing with a user.

The agent should be able to:

- Understand the user
- Formulate an appropriate response to a user utterance or query
- Conduct a conversation beyond simple voice commands or question answering
- Access world or domain knowledge, while keeping track of new developments

Consider what world knowledge is needed to answer these questions:

- ‘Siri, who won last night’s game?’
  - What day is today?   What teams won yesterday? What sports does the user follow? …
- ‘Alexa, what’s a moderately priced holiday present for my 10 year old niece?’
  - What holiday is it? What’s appropriate for a 10 year old girl? What’s moderately priced? …

Types of Dialogues and Dialogue Systems

Limited domain spoken dialogue systems

Ontology:

- a database that defines the properties of entities about which the dialogue system can converse

System-initiative vs user-initiative:

- who takes the initiative in the dialogue

System: Hello. Please tell me your date of birth using the six digit format.
System: Hello, how may I help you?

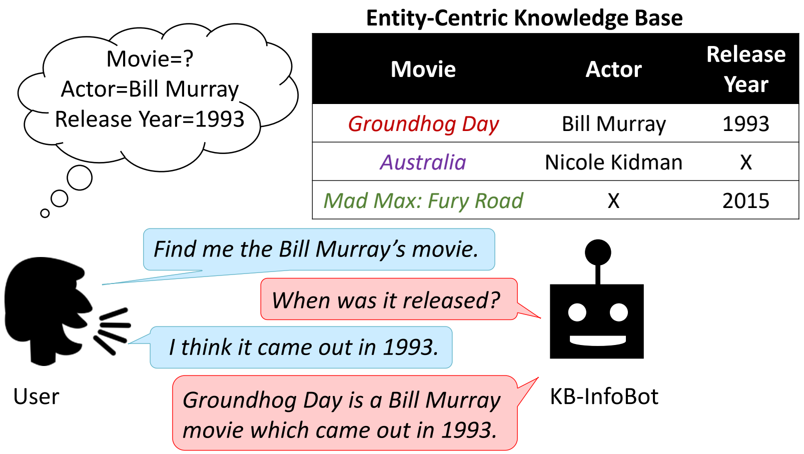
FYI: The GPT-2 Dialogue State Tracking Practical will focus on User-initiated dialogues

Types of Dialogue Systems

- **Question Answering**:
  - The agent should provide concise, direct answers to user queries based on rich knowledge drawn from various data sources including text collections such as Web documents and pre-compiled knowledge bases such as sales and marketing datasets
- **Task Completion / Task-Oriented Dialogue (TOD)**:
  - Designed to help users achieve pre-defined goals or tasks
  - Aim is to satisfy user in as few turns as possible
  - Agent interacts with knowledge bases, APIs, databases, etc.
    - Weather forecasts, Restaurant / Hotel / Train / Taxi bookings, …
- **Social Chat:**
  - Agent should converse seamlessly and appropriately - like a human as in the Turing test
  - Possible aims:
    - Simply to keep the user engaged
    - Provide useful recommendations
    - Monitor user behavior, e.g. for mental health assessment

QA/TOD : systems must be clear and correct – exactness matters!

Knowledge-Based Question Answering



Entity-Centric Knowledge Base

Movie	Actor	Release Year
Groundhog Day	Bill Murray	1993
Australia	Nicole Kidman	X
Mad Max: Fury Road	X	2015

Movie=?
Actor=Bill Murray
Release Year=1993

Find me the Bill Murray's movie.

When was it released?

I think it came out in 1993.

Groundhog Day is a Bill Murray movie which came out in 1993.

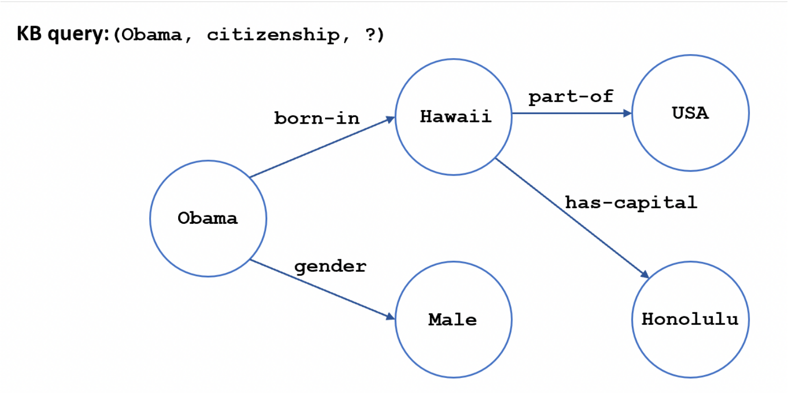
Multi-turn KB-QA

- System designed to ask only simple, easy-to-answer questions
- System interaction with user leads to KB query formulation

KB's are often in graph form, e.g. based on triples:

- (Obama, gender, male)
- (Hawaii, part-of, USA)

- Relations can be inferred from graphs , even if not explicit



KB query: `(Obama, citizenship, ?)`

Is President Obama a US citizen?

Question answering requires:

- Transforming a user question into a KB query
- Reasoning over the graph to find an answer

From Gao, Galley, Li

Social Chat – Rule-Based Systems

ELIZA

Responses are produced by Pattern/Transform rules

(* YOU * ME) -> (WHAT MAKES YOU THINK I (3) YOU)

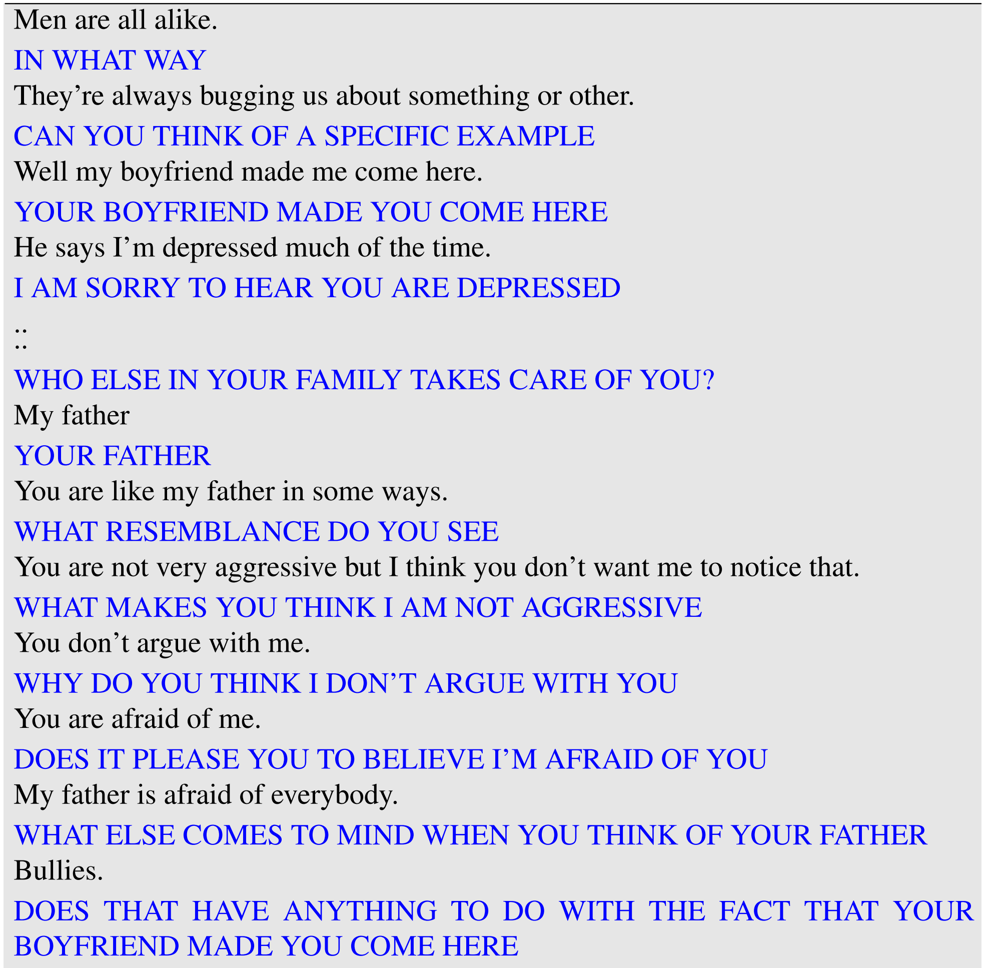
A pattern is identified in the user utterance, and a transform is applied to generate a system response.

The user utterance

You hate me

produces the response

WHAT MAKES YOU THINK I HATE YOU



Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
::
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?
My father
YOUR FATHER
You are like my father in some ways.
WHAT RESEMBLANCE DO YOU SEE
You are not very aggressive but I think you don't want me to notice that.
WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
You don't argue with me.
WHY DO YOU THINK I DON'T ARGUE WITH YOU
You are afraid of me.
DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU
My father is afraid of everybody.
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
Bullies.
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

from J&M 26.2

Social Chat - Corpus-Based Systems

Corpus-based chatbots exploit databases of human-human conversations

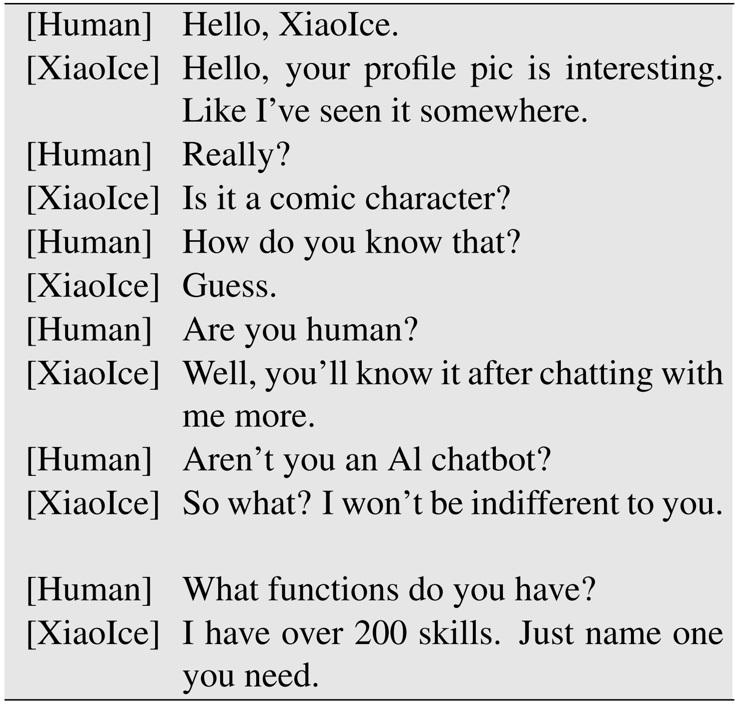
- Database consists of turns, which are pairs of query-and-response

Retrieval-based systems

- Respond to a user by repeating some appropriate turn from a natural text corpus

Generation-based systems

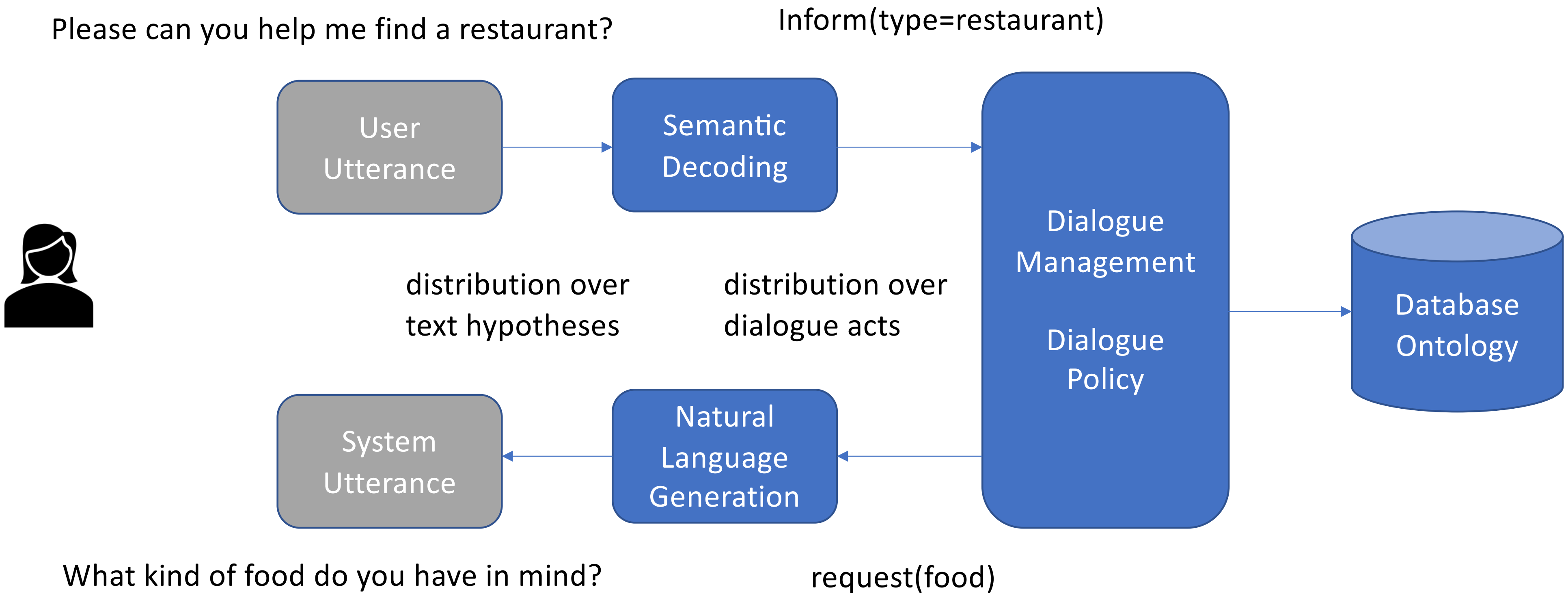
- Motivated by translation: system learns to generate responses from queries



[Human]	Hello, XiaoIce.
[XiaoIce]	Hello, your profile pic is interesting. Like I've seen it somewhere.
[Human]	Really?
[XiaoIce]	Is it a comic character?
[Human]	How do you know that?
[XiaoIce]	Guess.
[Human]	Are you human?
[XiaoIce]	Well, you'll know it after chatting with me more.
[Human]	Aren't you an Al chatbot?
[XiaoIce]	So what? I won't be indifferent to you.
[Human]	What functions do you have?
[XiaoIce]	I have over 200 skills. Just name one you need.

from J&M 26.2

Task Oriented Dialogue System Architecture



Please can you help me find a restaurant?

Inform(type=restaurant)

User Utterance

Semantic Decoding

Dialogue Management

Dialogue Policy

Database Ontology

distribution over text hypotheses

distribution over dialogue acts

System Utterance

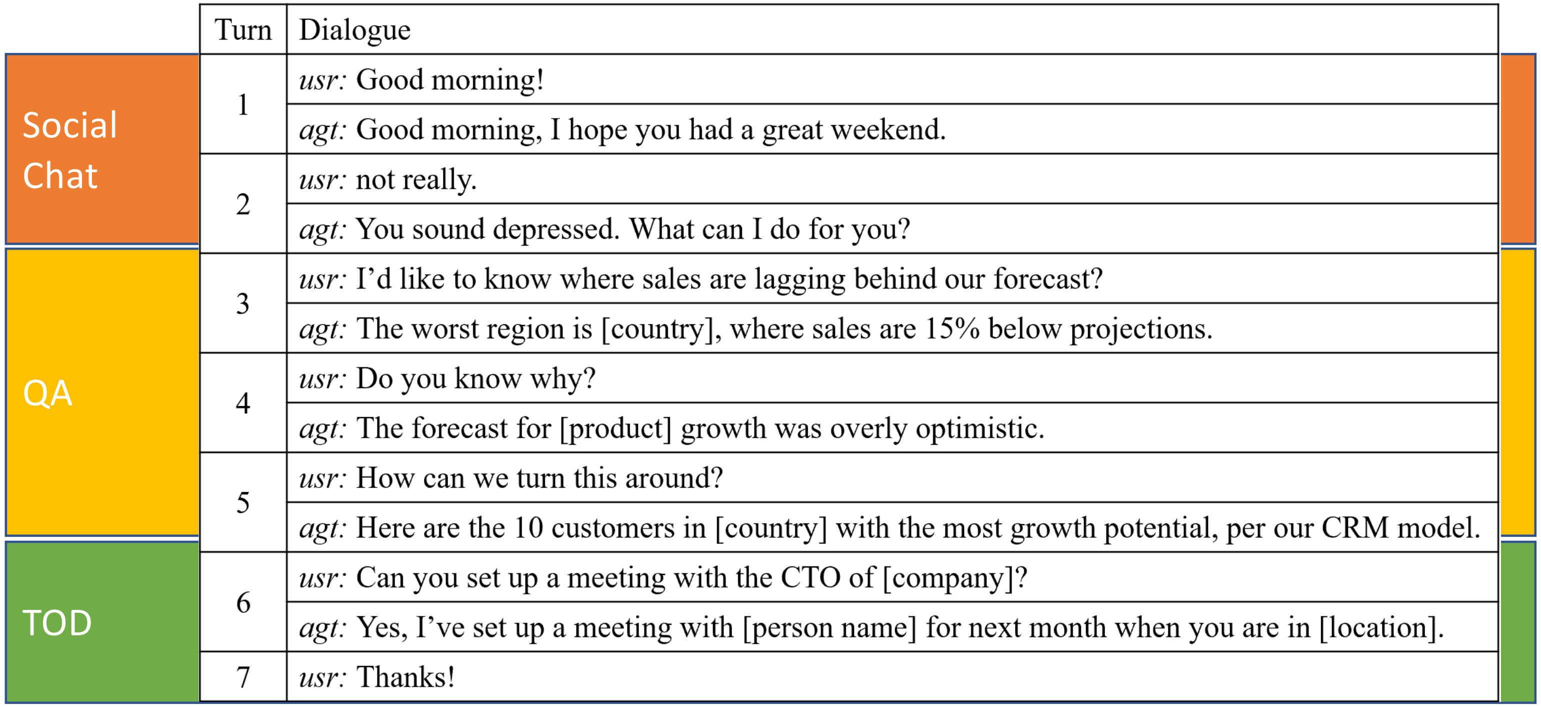
Natural Language Generation

What kind of food do you have in mind?

request(food)

UNIVERSITY OF CAMBRIDGE

Mixed Dialogues


Social Chat
QA
TOD

Turn	Dialogue
1	*usr:* Good morning!
	agt: Good morning, I hope you had a great weekend.
2	*usr:* not really.
	agt: You sound depressed. What can I do for you?
3	*usr:* I’d like to know where sales are lagging behind our forecast?
	agt: The worst region is [country], where sales are 15% below projections.
4	*usr:* Do you know why?
	agt: The forecast for [product] growth was overly optimistic.
5	*usr:* How can we turn this around?
	agt: Here are the 10 customers in [country] with the most growth potential, per our CRM model.
6	*usr:* Can you set up a meeting with the CTO of [company]?
	agt: Yes, I’ve set up a meeting with [person name] for next month when you are in [location].
7	*usr:* Thanks!

A human-agent dialogue during the process of making a business decision.

From Gao, Galley, Li

Task-Oriented vs Data-Driven Dialogue Systems

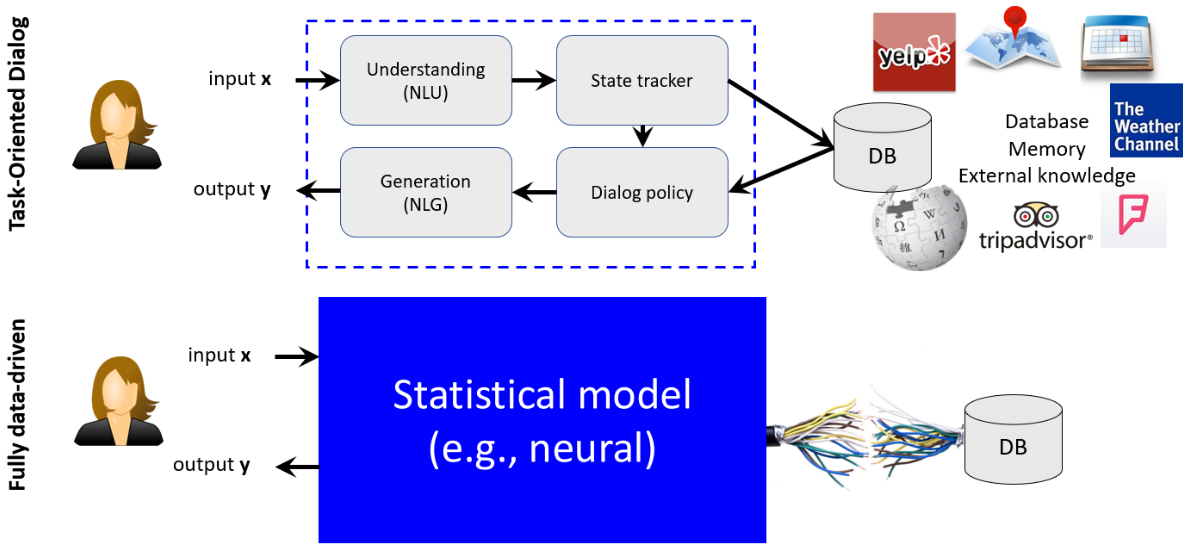


Figure 1.2: Two architectures of dialogue systems for (Top) traditional task-oriented dialogue and (Bottom) fully data-driven dialogue.

From Gao, Galley, Li

Characterizing Dialogues

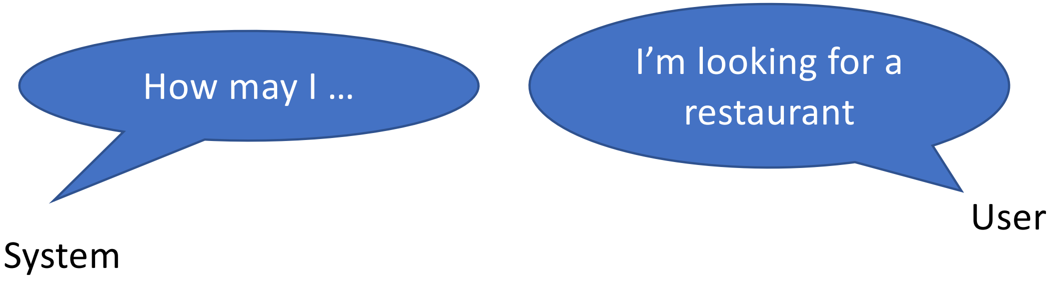
Turn-taking

Dialogue can be described in terms of system- and user-turns

- System: *How may I help you?*
- User: *I'm looking for a restaurant*
- System: *What kind of food would you like?*
- ….

Turn-taking can be more complex:

Barge-ins:



How may I …

System

I'm looking for a restaurant

User

Corrections**:** 'I said BAL-TI-MORE, not Boston'

Dialog Acts

A **dialogue act** represents the interactive function of a turn or a sentence

The question in this movie-booking example

“How many tickets do you need?”

is to request information about a certain slot:

request(num tickets)

The response

’’I want to watch it in Seattle.’’

is to inform the city name:

inform(city=“Seattle”)

System	Hello! How may I assist you?
User	Can I get tickets for Zoolander 2 tomorrow?
System	Sure! Which city?
User	I want to watch it in Seattle.

Dialogue Acts

A simple dialogue act formalism would consist of

Dialogue act type: encodes the system or user intention in dialogue turn

Semantic slots and values: ontology entries mentioned in the turn

Is there um maybe a cheap place in the centre of town please?

inform(price = cheap, area = centre)

dialogue act type

dialogue act type

Dialogue-State Architecture

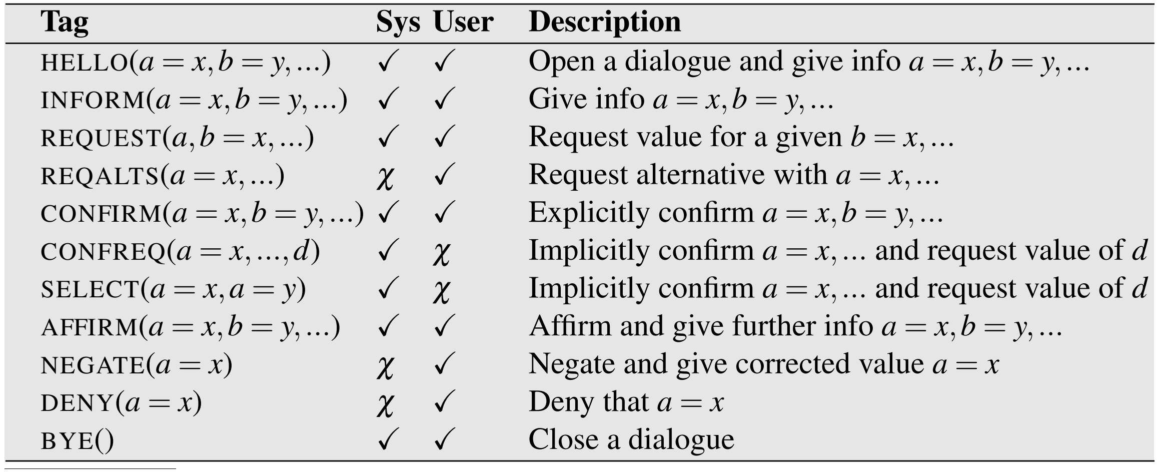
Slot-filling dialogues aim to identify **values** for a predefined set of **slots** to accomplish a task

- A task is finished when the requisite slots are filled with the correct values
- Slots depend on the **Domain**
    - Slots in the *Movie* domain: movie_name, theatre_name, time, date, ticket_price, ….
    - Values in the *Movie* domain correspond to movies currently showing
- Slot-Value pairs can be extracted from utterances
    - User: `For two people, please`   -> people_number=2

slot

value

Example - HIS Dialogue Acts

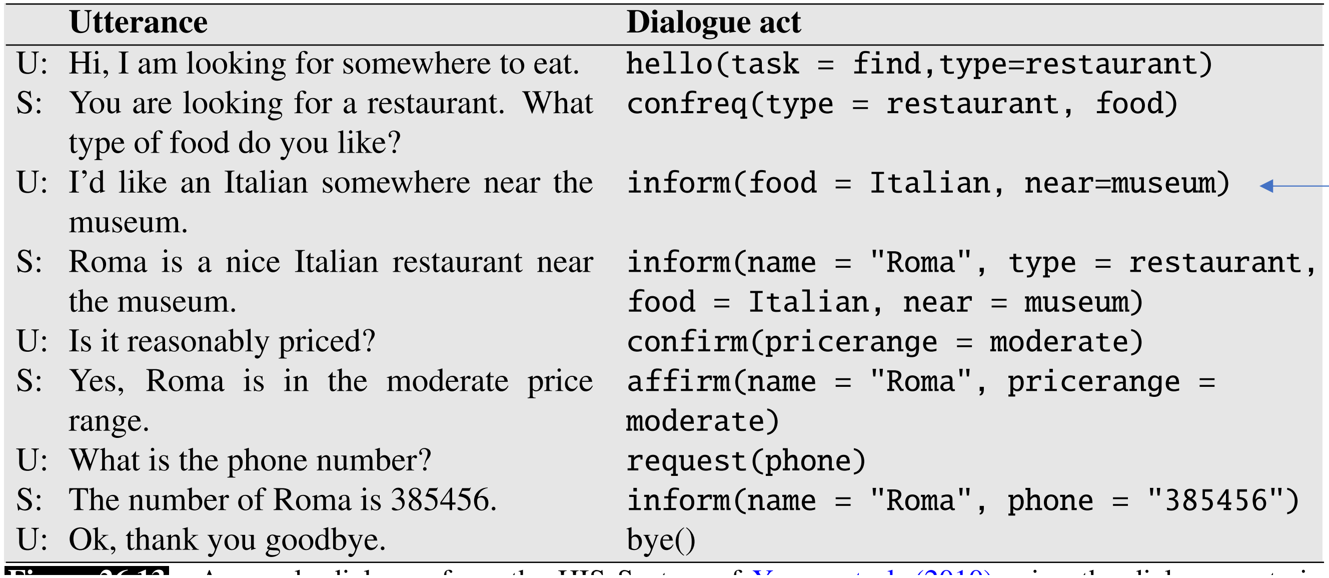


Tag	Sys	User	Description
HELLO$(a=x,b=y,\ldots)$	✓	✓	Open a dialogue and give info $a=x,b=y,\ldots$
INFORM$(a=x,b=y,\ldots)$	✓	✓	Give info $a=x,b=y,\ldots$
REQUEST$(a,b=x,\ldots)$	✓	✓	Request value for a given $b=x,\ldots$
REQALTS$(a=x,\ldots)$	χ	✓	Request alternative with $a=x,\ldots$
CONFIRM$(a=x,b=y,\ldots)$	✓	✓	Explicitly confirm $a=x,b=y,\ldots$
CONFREQ$(a=x,\ldots,d)$	✓	χ	Implicitly confirm $a=x,\ldots$ and request value of d
SELECT$(a=x,a=y)$	✓	χ	Implicitly confirm $a=x,\ldots$ and request value of d
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

- Both the System and the User can say \`Hello\` and \`Bye\`
- Only the User can \`Deny\`
- Only the System can implicitly confirm one value in a request for another value

Hidden Information State, from JM 26.12. Young et al. CSL. 2010

Example - HIS Sample Dialogue



	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()

The User dialogue act is 'Inform'
The act has two slot-value pairs:
1) Slot `food' has value 'Italian'
2) Slot `near' has value 'museum'

MultiWOZ Ontology
Dialogue Acts describe both user and system utterances

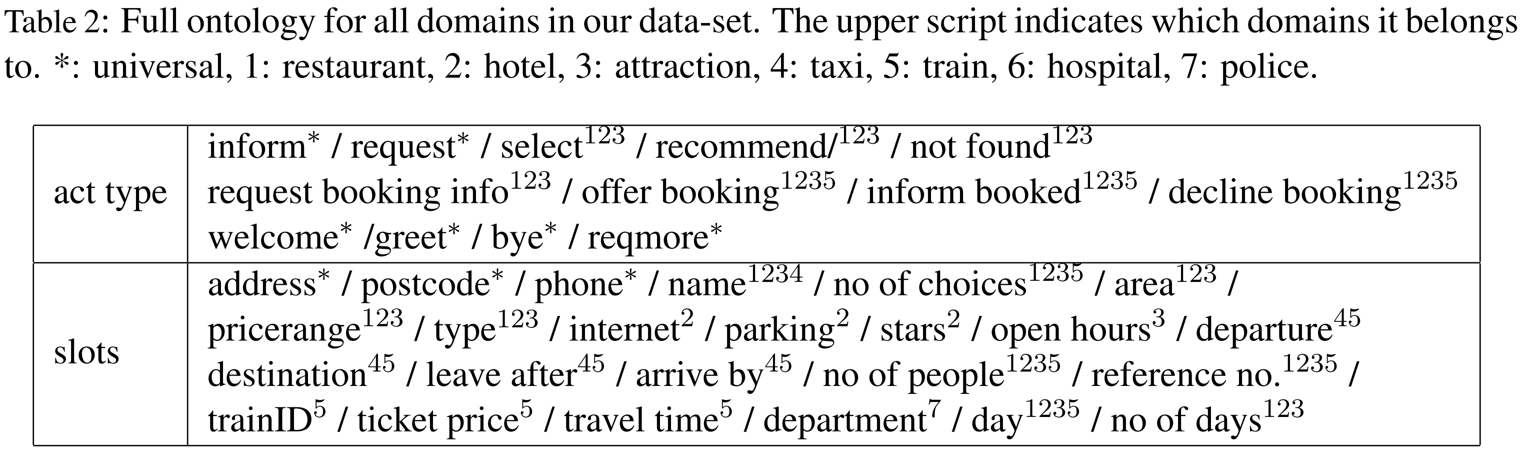
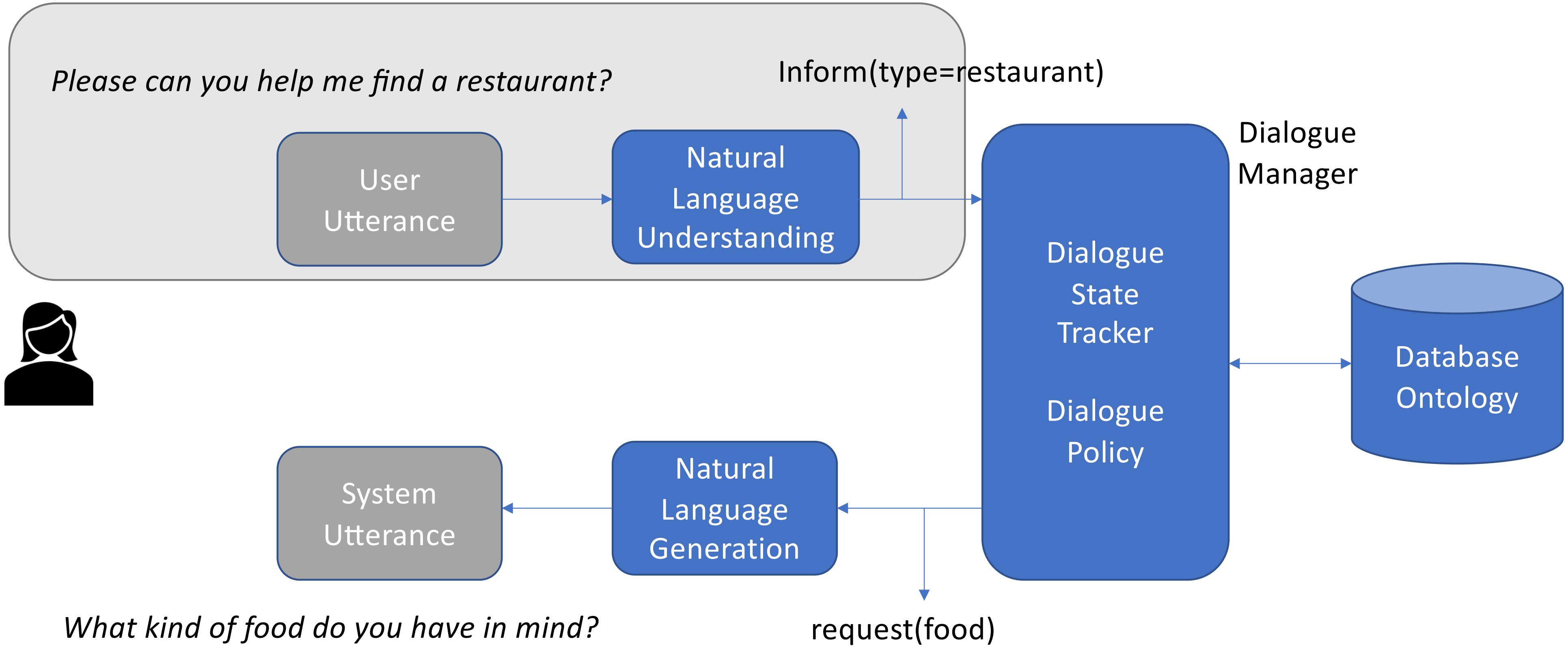


Table 2: Full ontology for all domains in our data-set. The upper script indicates which domains it belongs to. *: universal, 1: restaurant, 2: hotel, 3: attraction, 4: taxi, 5: train, 6: hospital, 7: police.

act type	inform* / request* / select^{123} / recommend/^{123} / not found^{123} request booking info^{123} / offer booking^{1235} / inform booked^{1235} / decline booking^{1235} welcome* /greet* / bye* / reqmore*
slots	address* / postcode* / phone* / name^{1234} / no of choices^{1235} / area^{123} / pricerange^{123} / type^{123} / internet^2 / parking^2 / stars^2 / open hours^3 / departure^{45} destination^{45} / leave after^{45} / arrive by^{45} / no of people^{1235} / reference no.^{1235} / trainID^5 / ticket price^5 / travel time^5 / department^7 / day^{1235} / no of days^{123}

Task Oriented Dialogue System Architecture

Probabilistic component models define distributions over text sequences and **dialogue acts**



Please can you help me find a restaurant?
Inform(type=restaurant)
User Utterance
Natural Language Understanding
Dialogue Manager
Dialogue State Tracker
Dialogue Policy
Database Ontology
System Utterance
Natural Language Generation
What kind of food do you have in mind?
request(food)

UNIVERSITY OF CAMBRIDGE

Natural Language Understanding

A *turn-level* analysis of a user utterance to extract:

- **Domain**
    - Does the utterance concern airlines, alarm clocks, movies, … ?
- **User intent**: What task or goal is the user trying to accomplish
    - Find a Movie, Get Flight Information, Book a Taxi, … ?
- **Slots & Values**
    - Extract all slot-value pairs from a sentence

Example NLU

Input: *Find recent comedies by James Cameron.*

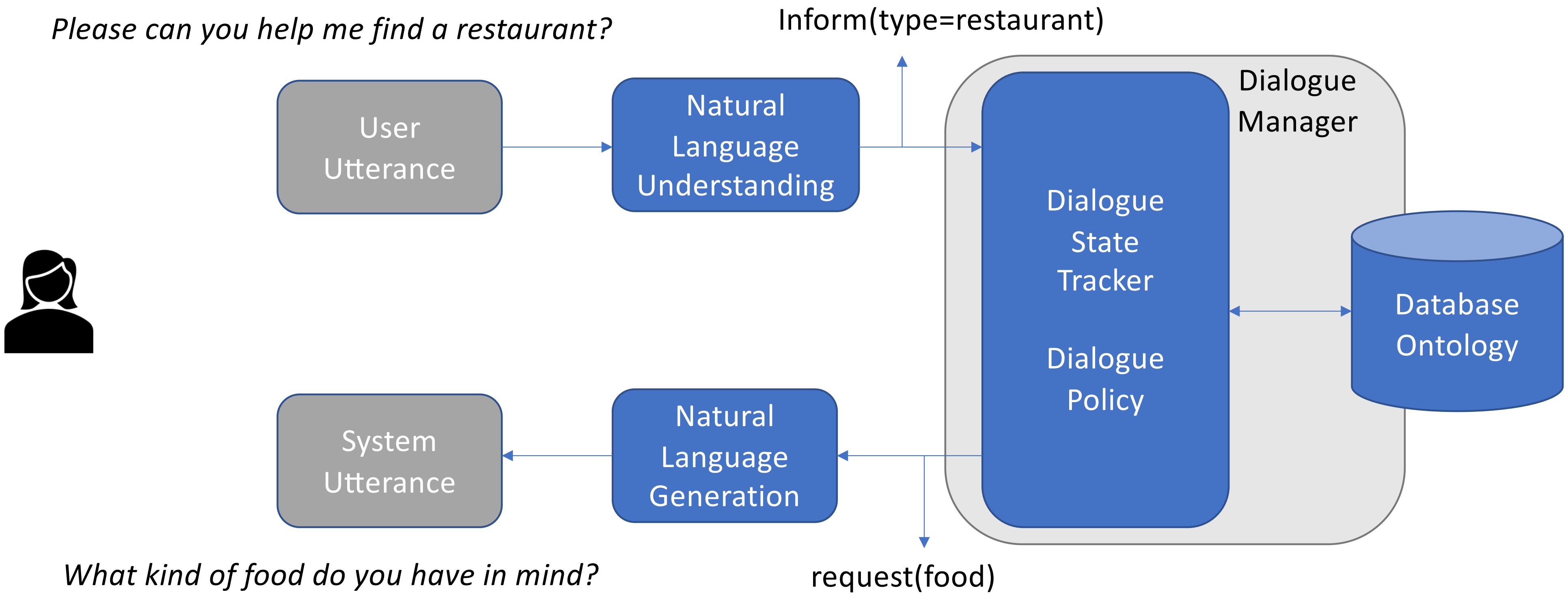
Output: **D:** movies

I: find_movie

S&Vs: date=recent, genre=comedies, director='James Cameron'

Task Oriented Dialogue System Architecture

Probabilistic component models define distributions over text sequences and **dialogue acts**



Please can you help me find a restaurant?

Inform(type=restaurant)

User Utterance

Natural Language Understanding

Dialogue Manager

Dialogue State Tracker

Dialogue Policy

Database Ontology

System Utterance

Natural Language Generation

What kind of food do you have in mind?

request(food)

Dialogue State Tracking (DST)

A *conversation level* analysis of the dialogue, including the current utterance

- In slot-filling problems, a dialogue state contains all information about what the user is looking for at the current turn of the conversation.
- This state is what the dialogue policy takes as input for deciding what action to take next
- The DST updates its state after each user utterance, possibly correcting earlier beliefs

<u>DST Example</u>

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

J&M 26

Dialogue Policy

The goal of the **dialogue policy** is to decide what action the system should take next, that is, what dialogue act to generate.

- The **Actions** the system is allowed to take are pre-defined
    - e.g. see the HIS dialogue act inventory
- **DP** decides the system action based on the dialogue state, as maintained by the DST
- **DP** can also access the database / knowledge-base

<u>Example DP</u>

Input:

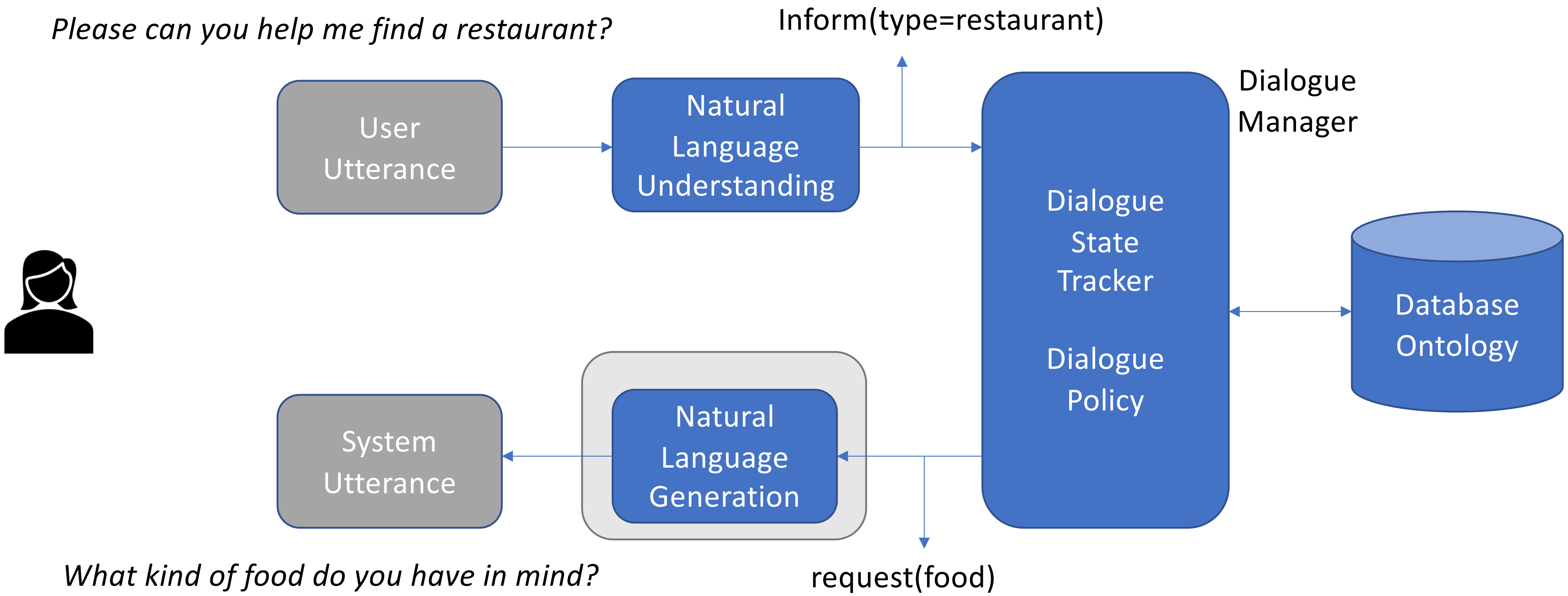
- Dialogue State: Hotel_Book(star=5, day=Sunday, people=2)
- KB State**:** res1=”Holiday Inn”

Output :

- System Action: inform(hotel_name=“Holiday Inn”)

Task Oriented Dialogue System Architecture

Probabilistic component models define distributions over text sequences and **dialogue acts**



Please can you help me find a restaurant?

User Utterance

Natural Language Understanding

Inform(type=restaurant)

Dialogue Manager

Dialogue State Tracker

Dialogue Policy

Database Ontology

Natural Language Generation

request(food)

System Utterance

What kind of food do you have in mind?

UNIVERSITY OF CAMBRIDGE

Natural Language Generation

NLG maps system actions to natural language responses

- Input: system speech-act , slot-values
- Output: natural language response

Example NLG

Input:

- System Action: inform(hotel_name=“Holiday Inn”)

Output:

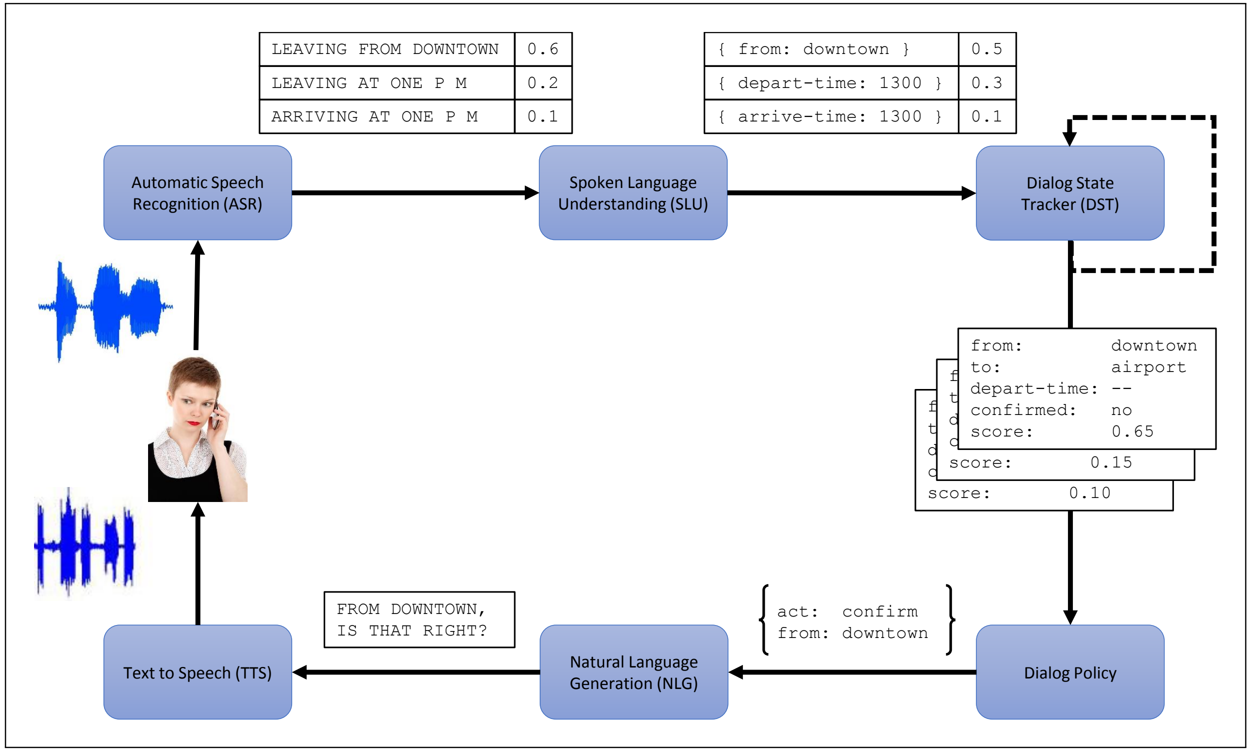
- System Response: ’I have booked the Holiday Inn for you’

NLG can be *template-based*

- inform(name=$A, phone=$B) => ’I found $A and their phone number is $B’

NLG can also be based on sequence-to-sequence models

A Task-Oriented Dialogue System, with ASR and 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

Automatic Speech Recognition (ASR)

Spoken Language Understanding (SLU)

Dialog State Tracker (DST)

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

```
score:         0.15
```

```
score:         0.10
```

Dialog Policy

```
act:   confirm
from:  downtown
```

Natural Language Generation (NLG)

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

Text to Speech (TTS)

Figure 26.11 Architecture of a dialogue-state system for task-oriented dialogue from Williams et al. (2016).

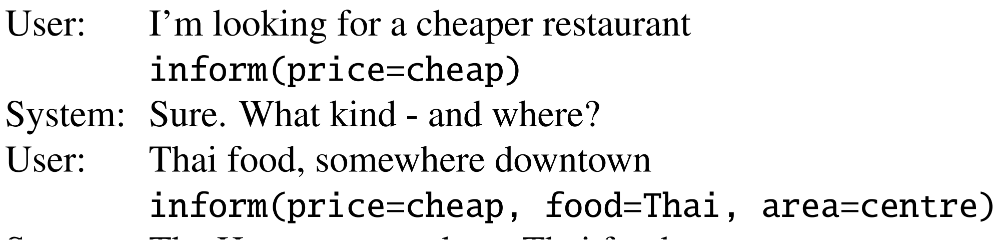
From J&M

Challenges in Dialogue Systems

- Common failure modes & deployment challenges
- Speech recognition errors
- Failure to keep track of the dialogue
- Wrong actions by the system
- Out-of-domain user requests

Linearization & Sequence-to-Sequence Models

Consider this dialogue snippet:



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)`

Linearisation maps *structures* such as *inform(price=cheap,food:Thai,area=centre)*

to *strings*: *inform price cheap <sep> inform food Thai <sep> inform area centre*

The component dialogue problems become string-to-string problems, for example:

- Natural Language Understanding:

  I’m looking for a cheaper restaurant -> inform price cheap

- Natural Language Generation

  act confirm <sep> from downtown -> From Downtown, is that right?

Summary

- Overview of Types of Dialogues and Dialogue Systems
    - Question Answering, Task Oriented Dialogue, Social Chat
- Task Oriented Dialogues
    - Ontologies
    - Component Problems
        - Natural Language Understanding
        - Dialogue State Tracking
        - Dialogue Policy
        - Natural Language Generation
- Linearisation