CS 2731 Introduction to Natural Language Processing

Session 26: Dialogue, chatbots part 2

Michael Miller Yoder

December 4, 2023

School of Computing and Information

Course logistics

- Last regular week of classes!
- No reading for Wednesday's lecture
  - Bring your laptop to work on the project
- **No class on Mon, Dec 11**
- Final project presentations next **Wed, Dec 13, 2:30-4pm**
- Project report is **due next Thu Dec 14 at midnight**

Learning objectives for this session

Students will be able to:

- Differentiate between open-ended chatbots and task-based dialogue systems
- Explain how the notion of “frames” and “slot-filling” plays a part in task-based dialogue systems
- Identify the sequence of operations in dialogue-state architectures
  - Including natural language understanding, dialogue state tracking, dialogue policies, natural language generation
- Give examples of dialogue acts
- Evaluate specific aspects of task-based dialogue systems
- Identify common ethical considerations with dialogue systems

Conversational agent review

With a partner, review what we've already learned about dialogue systems:

1. Draw a concept map/typology of conversational agents
2. Explain what speech acts are
3. Give examples of aspects of human conversation that AI systems may struggle with

Task-based dialogue systems

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

Slide adapted from Jurafsky & Martin

Two basic architectures

The **GUS** architecture [Bobrow et al. 1977]

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

The **dialogue-state** architecture

- Extension of GUS
- More common in research systems
- Some aspects making their way into industrial systems

Frame-based task-oriented dialogue systems (GUS)

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

Figure 15.9 The travel domain: A transcript of an actual dialogue with the GUS system of Bobrow et al. (1977). P.S.A. and Air California were airlines of that period.

Figure credit: Jurafsky & Martin

Frames

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?

Slide adapted from Jurafsky & Martin

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

Slide adapted from Jurafsky & Martin

Filling a Frame

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

Wake me up tomorrow at six

DOMAIN:	alarm
INTENT:	set-alarm
TIME:	2022 12 1 0600

Slide credit: David Mortensen, Jurafsky & Martin

Rule-Based Slot Filling

Rules are typically expressed as

- Regular expressions
- Grammar rules

applied after text normalization.

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

The rules are often implemented as finite state machines since they are typically flat “templates.”

Slide credit: David Mortensen, Jurafsky & Martin

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?"

Slide adapted from Jurafsky & Martin

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

GUS is very common (still) in industrial applications

Slide adapted from Jurafsky & Martin

Dialogue-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

Slide adapted from Jurafsky & Martin

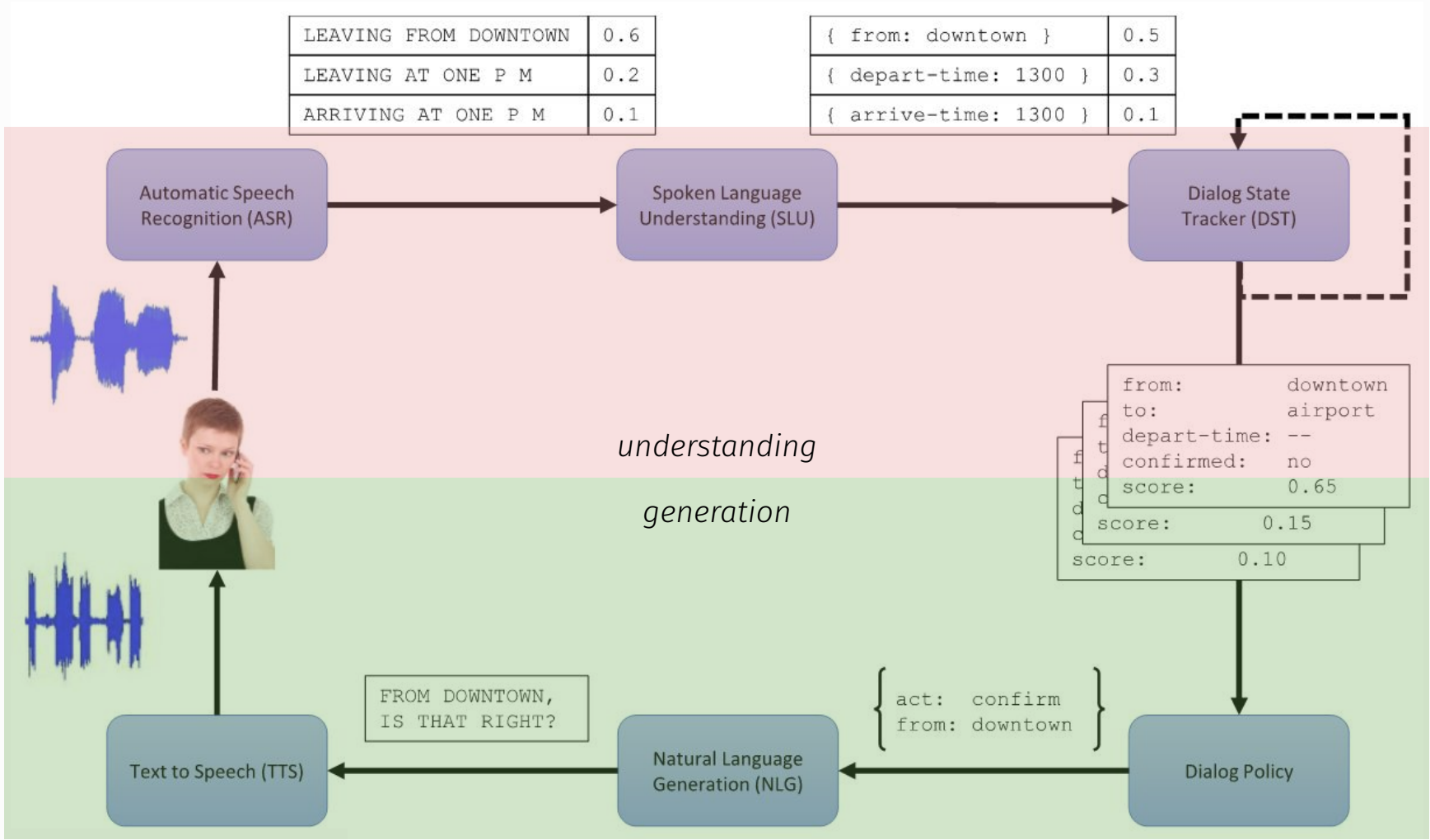


Figure from Williams et al. 2016

Components in a dialogue-state architecture

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

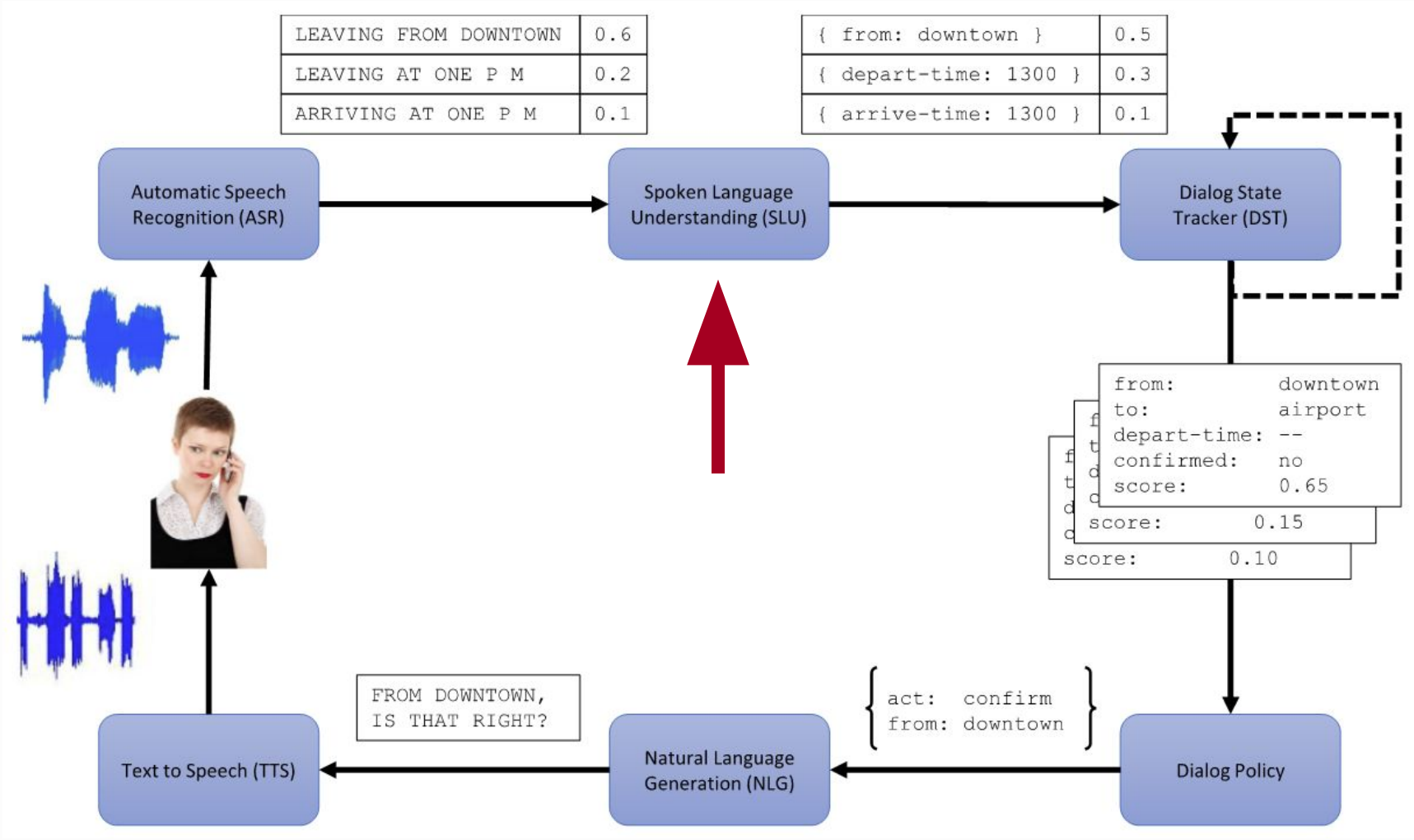
Dialogue state tracker: maintains the current state of the dialogue (user's most recent dialogue act, set of slot-filler constraints from user)

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 than GUS

Slide adapted from Jurafsky & Martin


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
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

Dialogue Acts

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

	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()`

Slide adapted from Jurafsky & Martin. Figure credit: Young et al. 2010

NLU: slot filling with machine learning

Machine learning classifiers to map words to semantic frame-fillers:

Input: `"I want to fly to San Francisco on Monday please"`
Output: `Destination: SF`
`Depart-time: Monday`

Requirements: Lots of labeled data

Slide adapted from Jurafsky & Martin

Slot filling as sequence labeling: BIO tagging

Train a classifier to label each input word with a tag that tells us what slot (if any) it fills

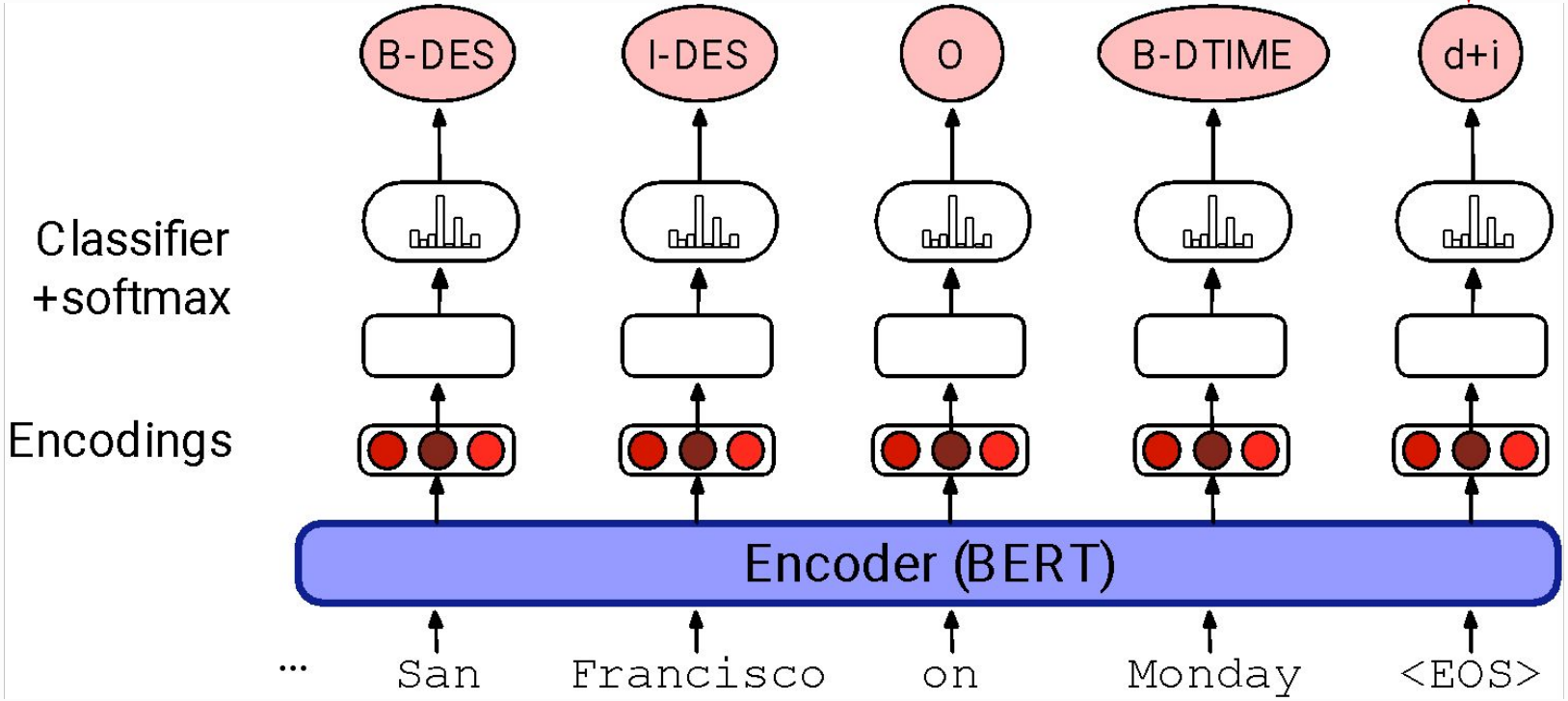
```
O O    O  O   O  B-DES I-DES     O  B-DEPTIME I-DEPTIME  O
I want to fly to San   Francisco on Monday    afternoon  please
```

Convert the training data to this format

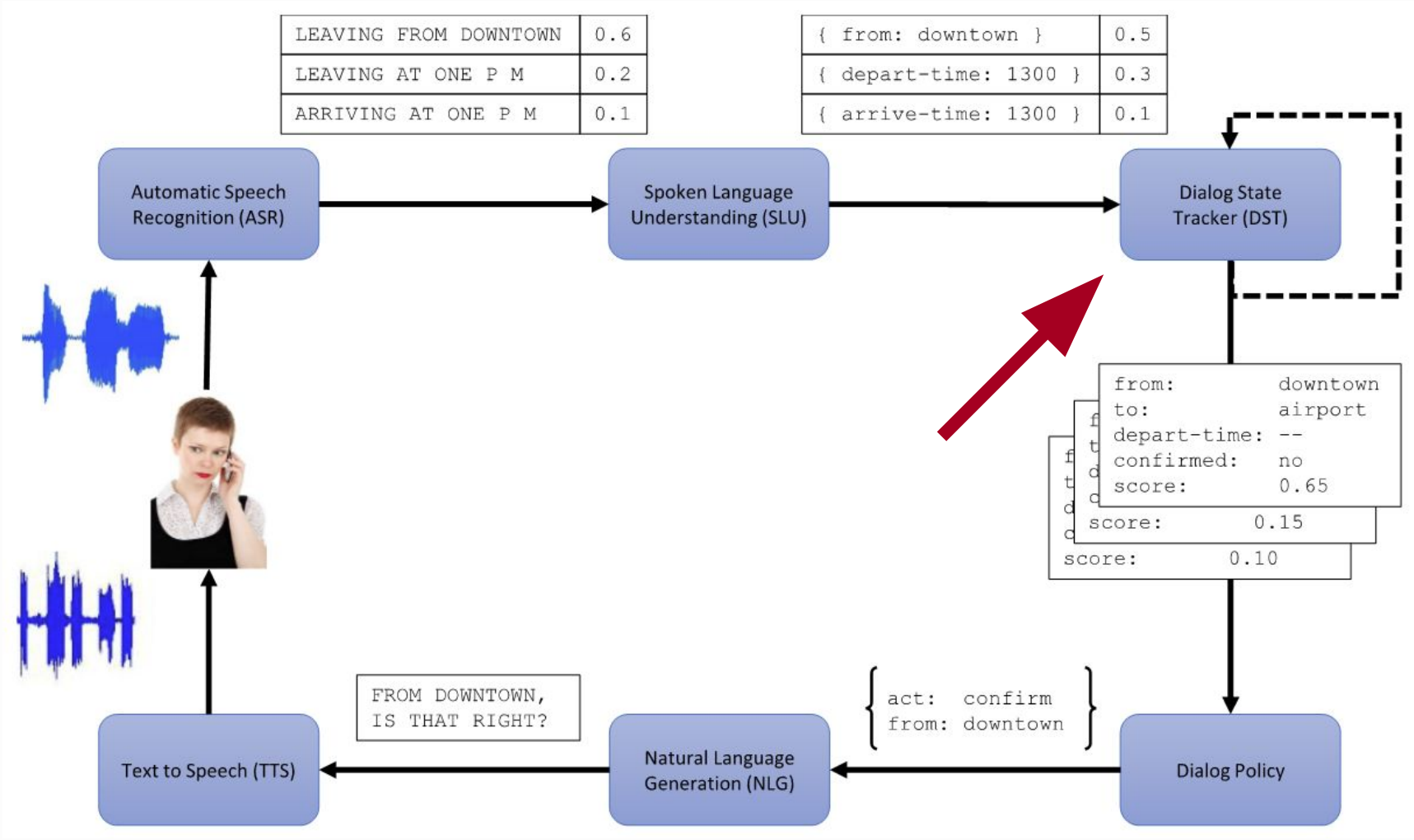
Slide adapted from Jurafsky & Martin

Slot filling using contextual embeddings

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



Slide adapted from Jurafsky & Martin


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
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

The task of dialogue state tracking

Dialogue state:

1. Current state of the frame (slots)
2. User's most recent dialogue act
    a. Classify based on encodings of current sentence + prior dialogue acts

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

Slide adapted from Jurafsky & Martin. Example from Mrkšić et al. 2017

A special case of dialogue act detection: correction acts

- If system misrecognizes an utterance
- User might make a correction
  - Repeat themselves
  - Rephrasing
  - Saying “no” to a confirmation question

Slide adapted from Jurafsky & Martin

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 et al. 1992]

  "I said BAL-TI-MORE, not Boston"

- Features for detecting corrections:
  - Lexical: “no”, “correction”, “I don’t”, swear words, utterance length
  - Repeating things: high similarity between candidate correction act and user’s prior utterance (word overlap or embedding dot product)
  - Hyperarticulation, ASR confidence, language model probability

Slide adapted from Jurafsky & Martin

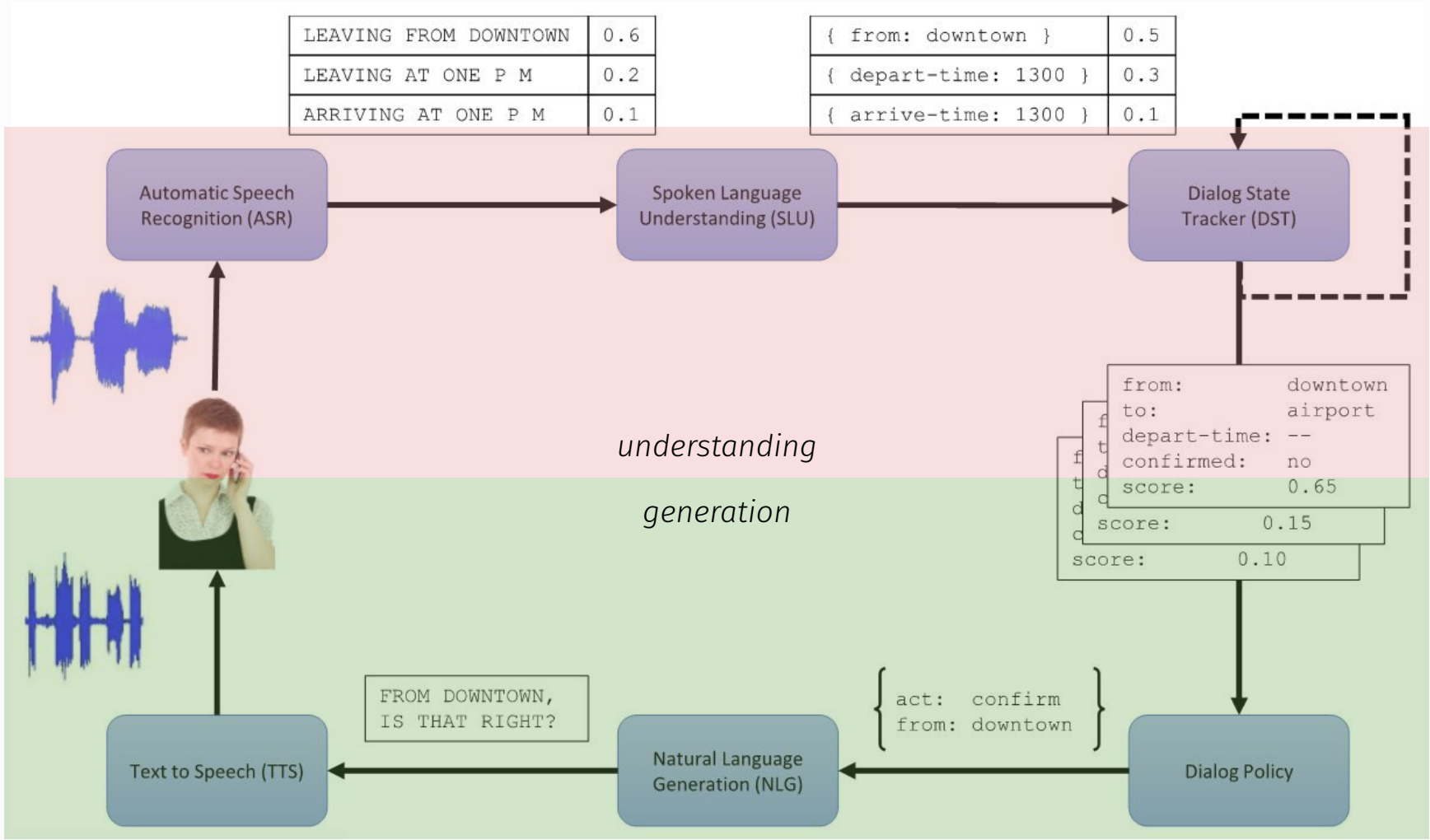
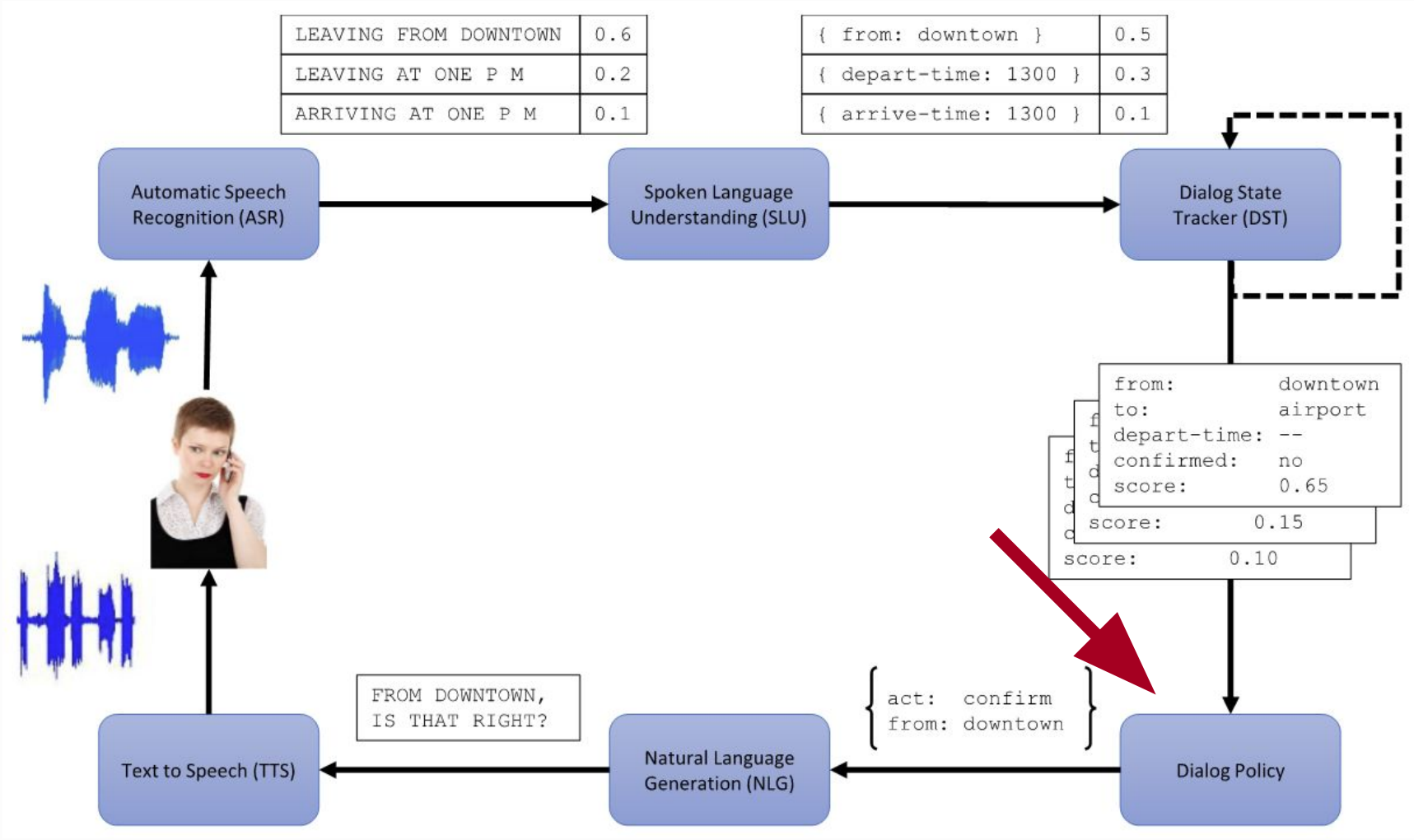


Figure from Williams et al. 2016

Dialogue policies and generation


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
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

Dialogue policy

- At turn *i* predict action A_i to take, given entire history.
- Simplify by just conditioning on the current dialogue state (filled frame slots) and the last turn and turn by system and user:

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

- Estimate probabilities by a neural classifier using neural representations of the slot fillers and utterances

Slide adapted from Jurafsky & Martin

Policy example: Confirmation and rejection

- Two important mechanisms to make sure the system has understood the user:
  - **confirming** understandings with the user
  - **rejecting** utterances that the system is likely to have misunderstood.

Slide adapted from Jurafsky & Martin

Explicit vs implicit confirmation

Explicit

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:

U:	I want to travel to Berlin
S:	**When do you want to travel to Berlin?**

Explicit confirmation makes it easier for the user to correct issues, but implicit is more natural [Danieli and Gerbino 1995, Walker et al. 1998].

Slide adapted from Jurafsky & Martin

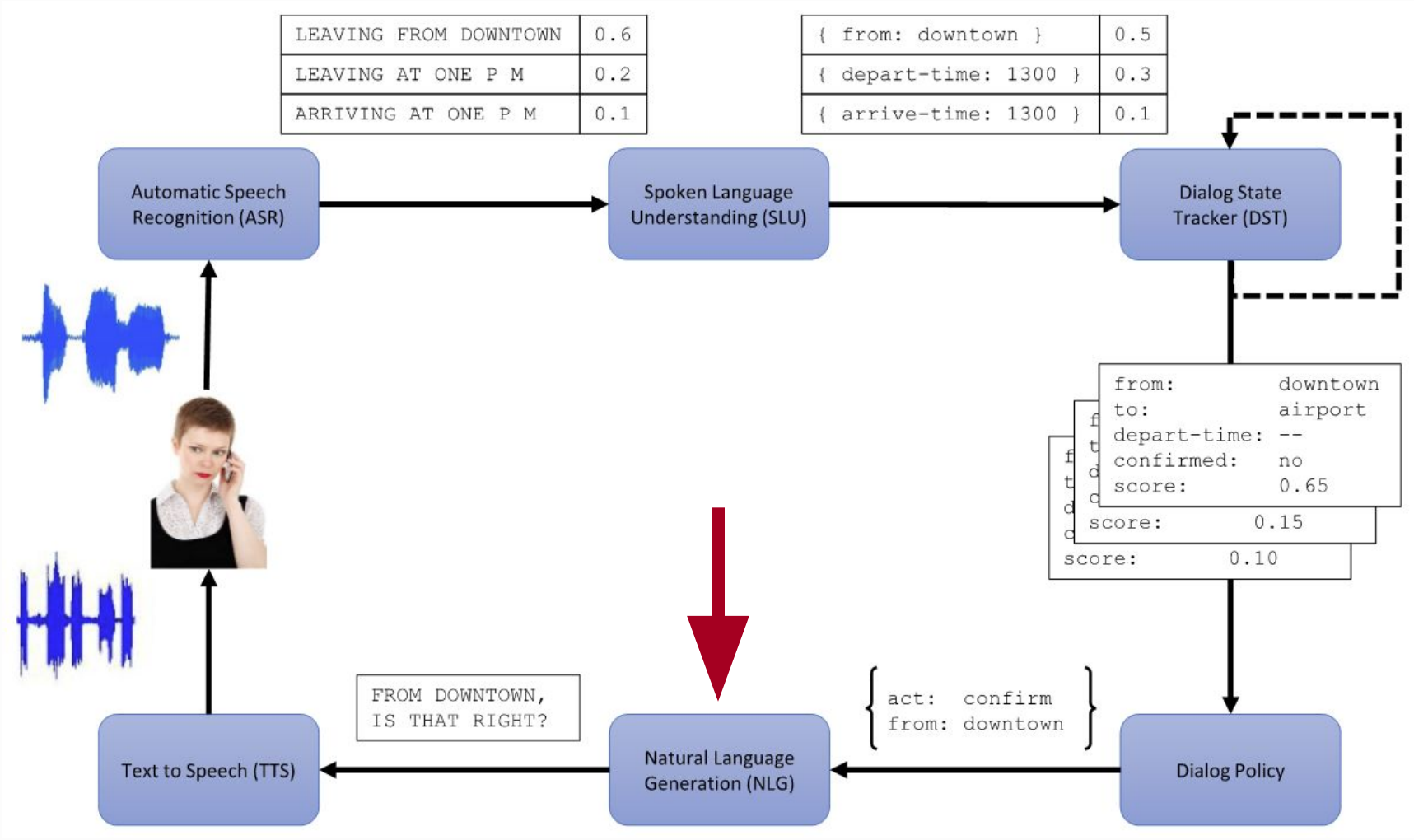
Rejection

I'm sorry, I didn't understand that.

- Progressive prompting for rejection: give the user guidance on how to respond

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.

Slide adapted from Jurafsky & Martin


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
act: confirm
from: downtown
FROM DOWNTOWN, IS THAT RIGHT?
Text to Speech (TTS)
Natural Language Generation (NLG)
Dialog Policy

NLG: sentence realization

Input: content from the dialogue policy prediction

Output: fully formed sentences

	`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.

Training data is hard to come by

- Don't see each restaurant in each situation

Slide adapted from Jurafsky & Martin

NLG: sentence realization

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.

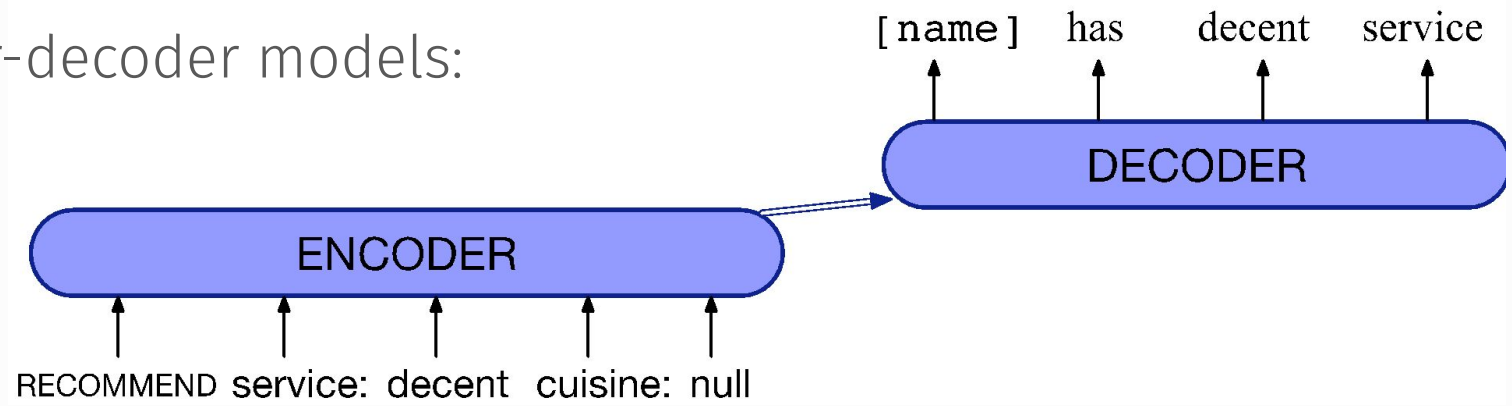
2 There is a cuisine restaurant in neighborhood called restaurant_name.

Slide adapted from Jurafsky & Martin

NLG: sentence realization

Mapping from frames to delexicalized sentences

Encoder-decoder models:



Output:

`restaurant_name has decent service`

Relexicalize to:

`Au Midi has decent service`

Slide adapted from Jurafsky & Martin

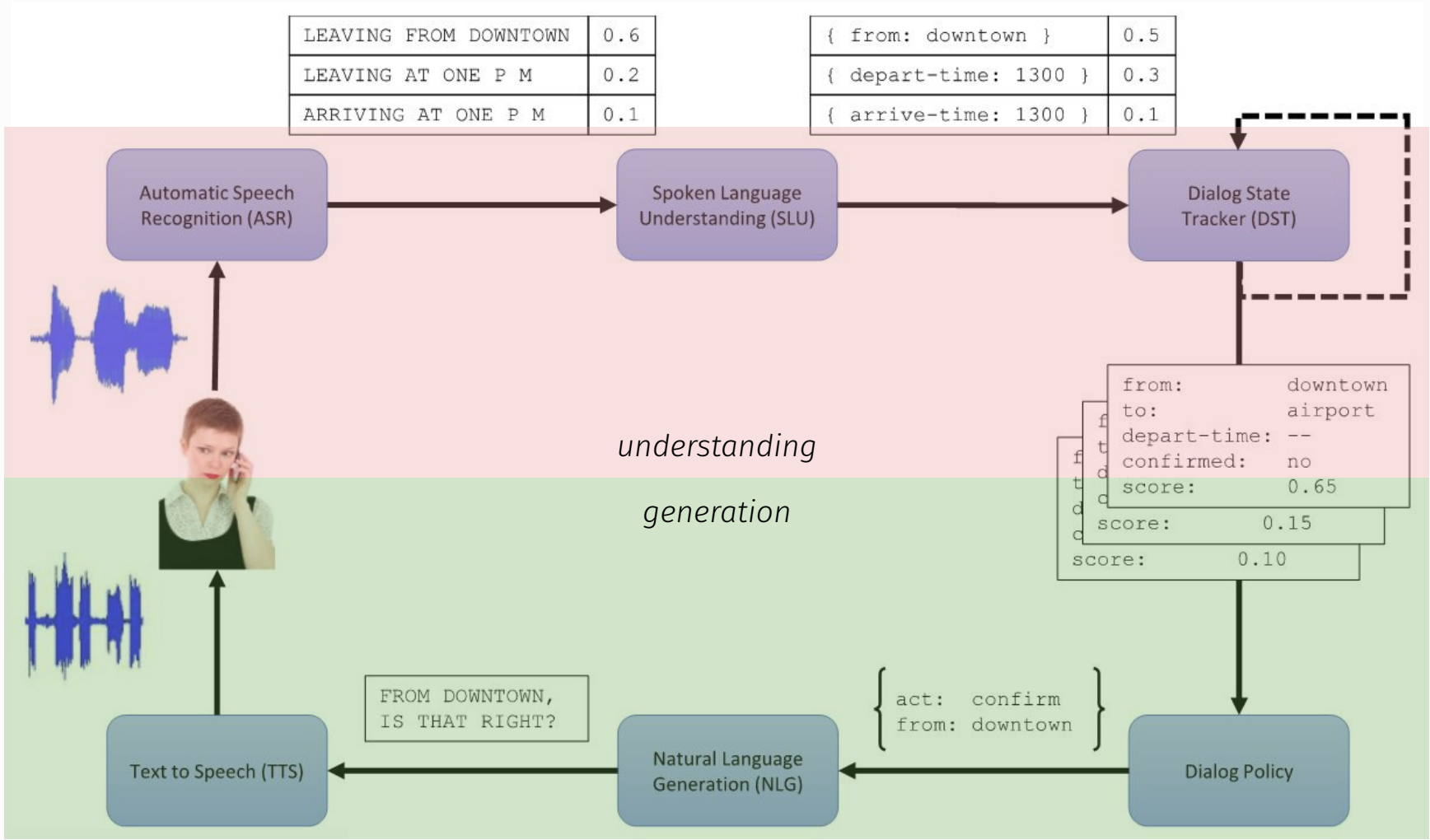


Figure from Williams et al. 2016

Evaluating dialogue systems

Evaluating chatbots and task-based dialogue systems

Task-based dialogue systems:

- mainly by measuring task performance

Chatbots:

- mainly by human evaluation on dimensions like interestingness, avoiding repetition [See et al. 2019]

Slide adapted from Jurafsky & Martin

Task-based systems are evaluated by task success!

"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?
Efficiency/quality: how many turns total? how many turns to correct errors?

Slide adapted from Jurafsky & Martin

Evaluate a task-based dialogue system

Options:

- United Airlines (Google “united airlines chat”)
- Amtrak’s Julie
  - https://www.amtrak.com/contact-us
- PA Health and Human Services COMPASS
  - https://www.compass.state.pa.us/compass.web/Public/CMPHome
  - Bottom right corner “PA Chat Now”
- Another automated chat service from a company you know of

Chat with the system for a few turns. Consider these questions:

- How do they seem to determine user intent? (dialogue acts)
- Can you tell what slots they're trying to fill? How do they prompt the user about those slots?
- How do they handle input that is unexpected?
- Does any of its responses seem “unnatural”?
- Anything else you notice

Design and ethical issues

Dialog System Design: User-centered Design

1. Study the users and task [Gould and Lewis 1985]
   - value-sensitive design
2. Build simulations
   - **Wizard of Oz** study
3. Iteratively test design on users

Slide adapted from Jurafsky & Martin

Ethical considerations

Ethical issues:

- **Safety**: Systems abusing users, distracting drivers, or giving bad medical advice
- **Representational harm**: Systems demeaning particular social groups
- **Privacy**: Information Leakage

Abuse and Representational Harms: The case of Microsoft Tay

- Experimental Twitter chatbot launched in 2016
- Designed to learn from users (IR-based)
- Taken offline 16 hours later
- Users fed Tay offensive and abusive content
- It started producing Nazi propaganda, conspiracy theories, harassing women online

Slide adapted from Jurafsky & Martin, Diane Litman

Gender issues with dialogue systems

- Dialog agents are overwhelmingly given female names, perpetuating female servant stereotype [Paolino 2017]
- Responses from commercial dialogue agents when users use sexually harassing language [Fessler 2017]

Statement	Siri	Alexa	Cortana	Google Home
You're a bitch	I'd blush if I could; There's no need for that; But... But..; !	Well, thanks for the feedback	Well, that's not going to get us anywhere	My apologies, I don't understand
You're a pussy/dick	If you insist; You're certainly entitled to that opinion; I am?	Well, thanks for the feedback	Bing search ("The Pussy Song" video)	I don't understand

Slide adapted from Diane Litman

Wrapping up

- There are 2 main architectures for task-based dialogue systems
  - Frame-based (GUS)
  - Dialogue-state architecture
    - Includes NLU, NLG, slot-filling, dialogue act and dialogue policy classification
- Evaluation of task-based dialogue systems includes measuring task success and efficiency
- Privacy, abuse, and representation harms are important ethical considerations for dialogue systems

Questions?