# Data for Task-Oriented Dialogue

## Data

- How do you get training data when you don't have a working dialogue system?
- Somehow need to annotate what should happen in response to user utterances. But then you need to know how those users would respond…

## Full Dialogue Task

Find me a good sushi restaurant in Chelsea

```
Intent=restaurant
restaurant_type <- sushi
location <- Chelsea
curr_result <- execute_search()
```

Sushi Seki Chelsea is a sushi restaurant in Chelsea with 4.4 stars on Google

How expensive is it?

```
get_value(cost, curr_result)
```

Entrees are around $30 each

## Reward for completing task?

Find me a good sushi restaurant in Chelsea

```
Intent=restaurant
restaurant_type <- sushi
location <- Chelsea
curr_result <- execute_search()
```

Very indirect signal of what should happen up here

Sushi Seki Chelsea is a sushi restaurant in Chelsea with 4.4 stars on Google

How expensive is it?

…

Okay make me a reservation!

**+1** `make_reservation(curr_result)`

# User gives reward?

Find me a good sushi restaurant in Chelsea

```
Intent=restaurant
restaurant_type <- sushi
location <- Chelsea
curr_result <- execute_search()
```

How does the user know the right search happened?

**+1** Sushi Seki Chelsea is a sushi restaurant in Chelsea with 4.4 stars on Google

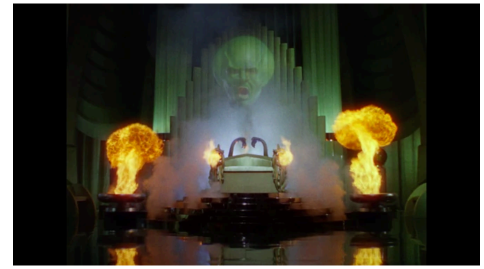
How expensive is it?

```
get_value(cost, curr_result)
```

**+1** Entrees are around $30 each

# Wizard-of-Oz

- Learning from demonstrations: "wizard" pulls the levers and makes the dialogue system update its state and take actions



Kelley (early 1980s), Ford and Smith (1982)

# Full Dialogue Task

Find me a good sushi restaurant in Chelsea

wizard enters these

```
Intent=restaurant
restaurant_type <- sushi
location <- Chelsea
curr_result <- execute_search()
```

wizard types this out or invokes templates

Sushi Seki Chelsea is a sushi restaurant in Chelsea with 4.4 stars on Google

- Wizard can be a trained expert and know exactly what the dialogue systems is supposed to do

# Full Dialogue Task

Find me a good sushi restaurant in Chelsea

```
Intent=restaurant
restaurant_type <- sushi
location <- Chelsea
stars <- 4+
curr_result <- execute_search()
```

- User asked for a "good" restaurant — does that mean we should filter by star rating? What does "good" mean?
- Hard to change system behavior if training from static traces, especially if system capabilities or desired behavior change

# ATIS

# Air Travel Information Service (ATIS)

- Given an utterance, predict a domain-specific semantic interpretation

| Utterance | *How much is the cheapest flight from Boston to New York tomorrow morning?* |
|---|---|
| Goal: | Airfare |
| Cost_Relative | *cheapest* |
| Depart_City | *Boston* |
| Arrival_City | *New York* |
| Depart_Date.Relative | *tomorrow* |
| Depart_Time.Period | *morning* |

DARPA (early 1990s), Figure from Tur et al. (2010)

# Intents

- 29 different intents

which flights go from cleveland to indianapolis on april fifth

**Intent**: flight

does tacoma airport offer transportation from the airport to the downtown area

**Intent**: ground_service

what days of the week do flights from san jose to nashville fly on

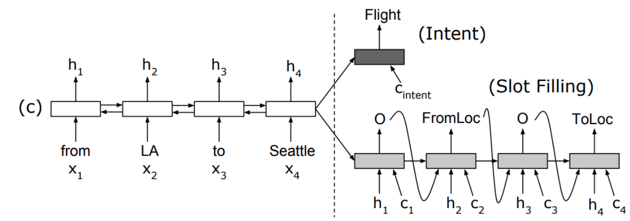
**Intent**: day_name

what meals are served on american flight 811 from tampa to milwaukee

**Intent**: meal

# Joint Intent Classification and Tagging

- RNN jointly predicts intent and slot tags




| Model | F1 Score | Intent Error (%) |
|---|---|---|
| RecNN [8] | 93.22 | 4.60 |
| RecNN+Viterbi [8] | 93.96 | 4.60 |
| Attention Encoder-Decoder NN (with aligned inputs) | **95.87** | **1.57** |
| Attention BiRNN | **95.98** | **1.79** |

Liu and Lane (2016)

# Air Travel Information Service (ATIS)

$\bar{x}_1$: *show me flights from seattle to boston next monday*

$\bar{y}_1$:
```
(SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.from_airport IN (SELECT
airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT
city.city_code FROM city WHERE city.city_name = 'SEATTLE'))) AND (flight.to_airport IN (SELECT
airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT
city.city_code FROM city WHERE city.city_name = 'BOSTON'))) AND (flight.flight_days IN (SELECT
days.days_code FROM days WHERE days.day_name IN (SELECT date_day.day_name FROM date_day WHERE
date_day.year = 1993 AND date_day.month_number = 2 AND date_day.day_number = 8))));
```

$\bar{x}_2$: *on american airlines*

$\bar{y}_2$:
```
(SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.airline_code = 'AA') AND (flight.from_airport
IN (SELECT airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT
city.city_code FROM city WHERE city.city_name = 'SEATTLE'))) AND (flight.to_airport IN (SELECT air
port_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code
FROM city WHERE city.city_name = 'BOSTON'))) AND (flight.flight_days IN (SELECT days.days_code FROM
days WHERE days.day_name IN (SELECT date_day.day_name FROM date_day WHERE date_day.year = 1993 AND
date_day.month_number = 2 AND date_day.day_number = 8))));
```

- Need to use dialogue context to do the right thing. Here we're appending American Airlines as a constraint to the previous query
- seq2seq model mapping to query with copy mechanism

Suhr et al. (2018)

# Air Travel Information Service (ATIS)

- Detect and anonymize entities for better performance
- Common trick in QA models (discussed next week)

Original utterance and query:

$\bar{x}_1$: *show me flights from seattle to boston next monday*

$\bar{y}_1$:
```
( SELECT DISTINCT flight.flight_id ...
city.city_name = 'SEATTLE' ...  city.city_name
= 'BOSTON' ...  date_day.year = 1993 AND
date_day.month_number = 2 AND date_day.day_number
= 8 ...
```

Anonymized utterance and query:

$\bar{x}'_1$: *show me flights from CITY#1 to CITY#2 DAY#1 MONTH#1 YEAR#1*

$\bar{y}'_1$:
```
( SELECT DISTINCT flight.flight_id ...
city.city_name = CITY#1 ...  city.city_name
= CITY#2 ...  date_day.year = YEAR#1
AND date_day.month_number = MONTH#1 AND
date_day.day_number = DAY#1 ...
```

Anonymization mapping:

| | | | |
|---|---|---|---|
| CITY#1 | 'SEATTLE' | MONTH#1 | 2 |
| CITY#2 | 'BOSTON' | YEAR#1 | 1993 |
| DAY#1 | 8 | | |

Suhr et al. (2018)

# Air Travel Information Service (ATIS)

| **Model** | **Query** | **Denotation** | |
|---|---|---|---|
| | | **Relaxed** | **Strict** |
| **Development Results** | | | |
| SEQ2SEQ-0 | 28.7±1.7 | 48.8±1.4 | 43.2±1.8 |
| SEQ2SEQ-H | 35.1±2.2 | 59.4±2.4 | 56.7±3.2 |
| S2S+ANON | 37.6±0.7 | 61.6±0.7 | 60.6±0.7 |
| FULL-0 | 36.3±0.5 | 61.5±0.8 | 61.0±0.9 |
| FULL | 37.5±0.9 | 63.0±0.7 | **62.5**±0.9 |

- -H has access to previous utterances
- Denotation: right answer but not exact match on query

Suhr et al. (2018)

# Goal-oriented Dialogue

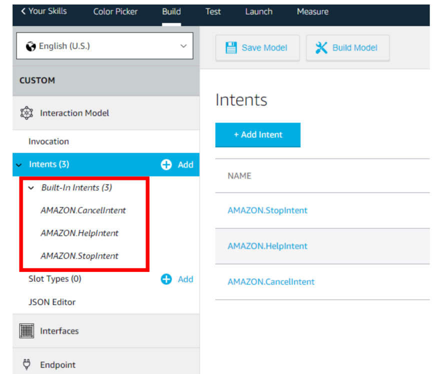
- Tons of industry interest!
- Dozens of startups + medium-sized companies in this space
- Big Companies: Apple Siri (VocalIQ), Google Allo, Amazon Alexa, Microsoft Cortana, Facebook M, Samsung Bixby, Tencent WeChat
- Lots of cool work that's not public yet

# Alexa Skills

- Let you add functionality to Amazon Alexa
- Can deploy to Alexa devices, develop and debug through AWS (no device necessary)
- Plugs into Amazon's ASR and TTS, so no need to wrestle with these services yourself

# Alexa Skills

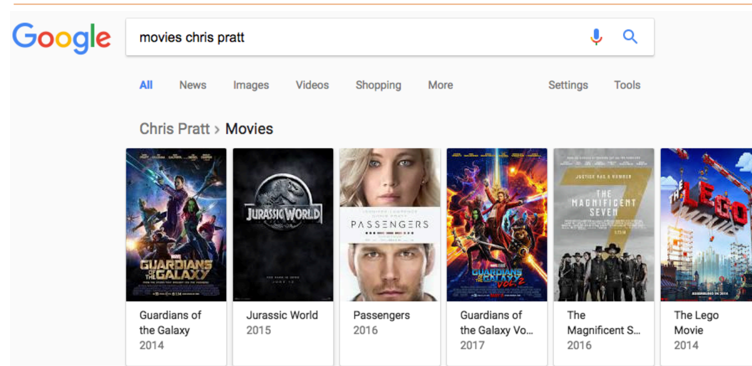
- Instantiate skill, intents, and slots
- Custom slot types let you define values
- To do really complicated things, need phrase slots — this actually gives you the text
- More information: EE596 from UW




Slide credit: Hao Fang / Hao Cheng (UW)

# Other Dialogue Applications

# Search/QA as Dialogue




- "Has Chris Pratt won an Oscar?" / "Has *he* won an Oscar"

# QA as Dialogue

- Dialogue is a very natural way to find information from a search engine or a QA system
- Challenges:
  - QA is hard enough on its own
  - Users move the goalposts

**Original intent:**
What super hero from Earth appeared most recently?

**1.** Who are all of the super heroes?

**2.** Which of them come from Earth?

**3.** Of those, who appeared most recently?

**Legion of Super Heroes Post-*Infinite Crisis***

| *Character* | *First Appeared* | *Home World* | *Powers* |
|---|---|---|---|
| Night Girl | 2007 | Kathoon | Super strength |
| Dragonwing | 2010 | Earth | Fire breath |
| Gates | 2009 | Vyrga | Teleporting |
| XS | 2009 | Aarok | Super speed |
| Harmonia | 2011 | Earth | Elemental |

Iyyer et al. (2017)

# QA as Dialogue

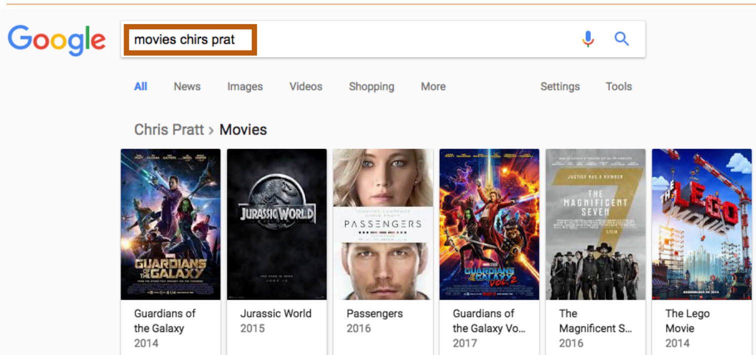
- UW QuAC dataset: Question Answering in Context

**Section:** Daffy Duck, Origin & History

STUDENT: **What is the origin of Daffy Duck?**
TEACHER: ↪ first appeared in Porky's Duck Hunt
STUDENT: **What was he like in that episode?**
TEACHER: ↪ assertive, unrestrained, combative
STUDENT: **Was he the star?**
TEACHER: ↪ No, barely more than an unnamed bit player in this short
STUDENT: **Who was the star?**
TEACHER: ↛ No answer
STUDENT: **Did he change a lot from that first episode in future episodes?**
TEACHER: ↪ Yes, the only aspects of the character that have remained consistent (...) are his voice characterization by Mel Blanc
STUDENT: **How has he changed?**
TEACHER: ↪ Daffy was less anthropomorphic
STUDENT: **In what other ways did he change?**
TEACHER: ↪ Daffy's slobbery, exaggerated lisp (...) is barely noticeable in the early cartoons.
STUDENT: **Why did they add the lisp?**
TEACHER: ↪ One often-repeated "official" story is that it was modeled after producer Leon Schlesinger's tendency to lisp.
STUDENT: **Is there an "unofficial" story?**
TEACHER: ↪ Yes, Mel Blanc (...) contradicts that conventional belief
...

Choi et al. (2018)

# Search as Dialogue

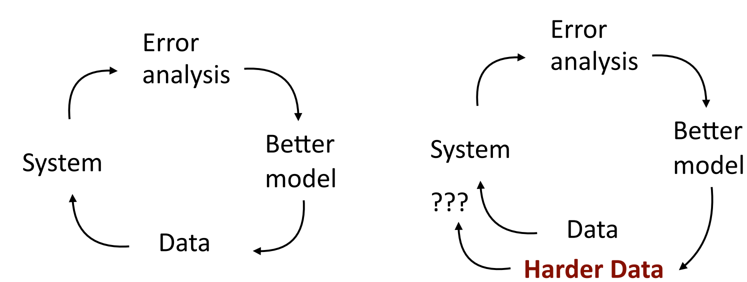



- Google can deal with misspellings, so more misspellings happen — Google has to do more!

# Dialogue Mission Creep

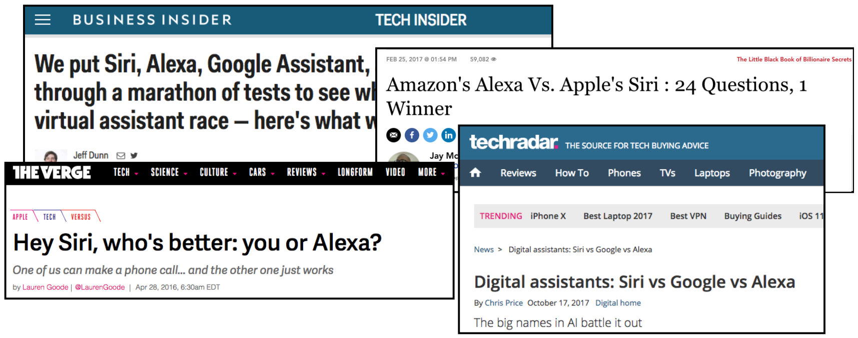
Most NLP tasks

Dialogue/Search/QA




- Fixed distribution (e.g., natural language sentences), error rate -> 0
- Error rate -> ???; "mission creep" from HCI element

# Dialogue Mission Creep




- High visibility — your product has to work really well!