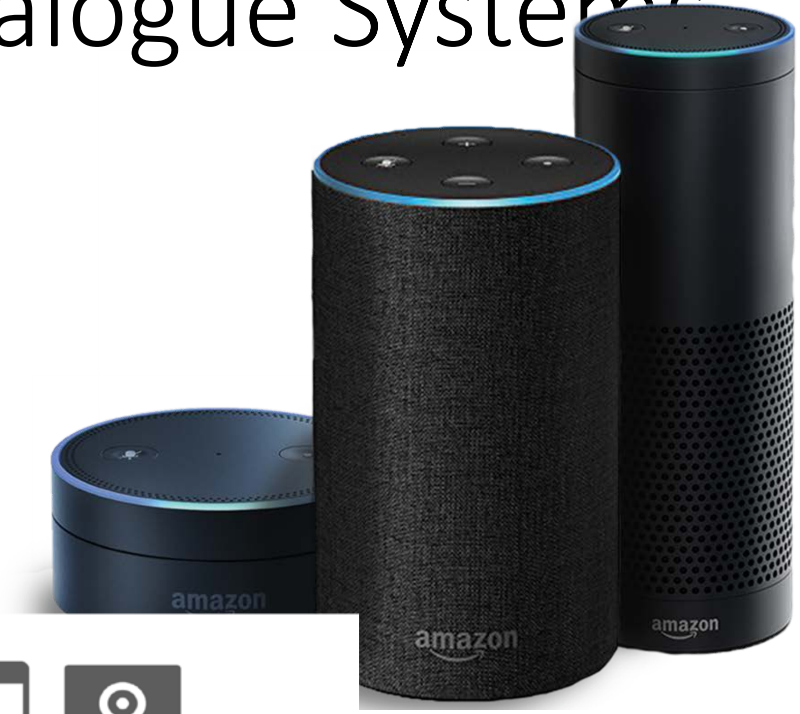
CS60075
Natural Language Processing
Autumn 2020

Module 10

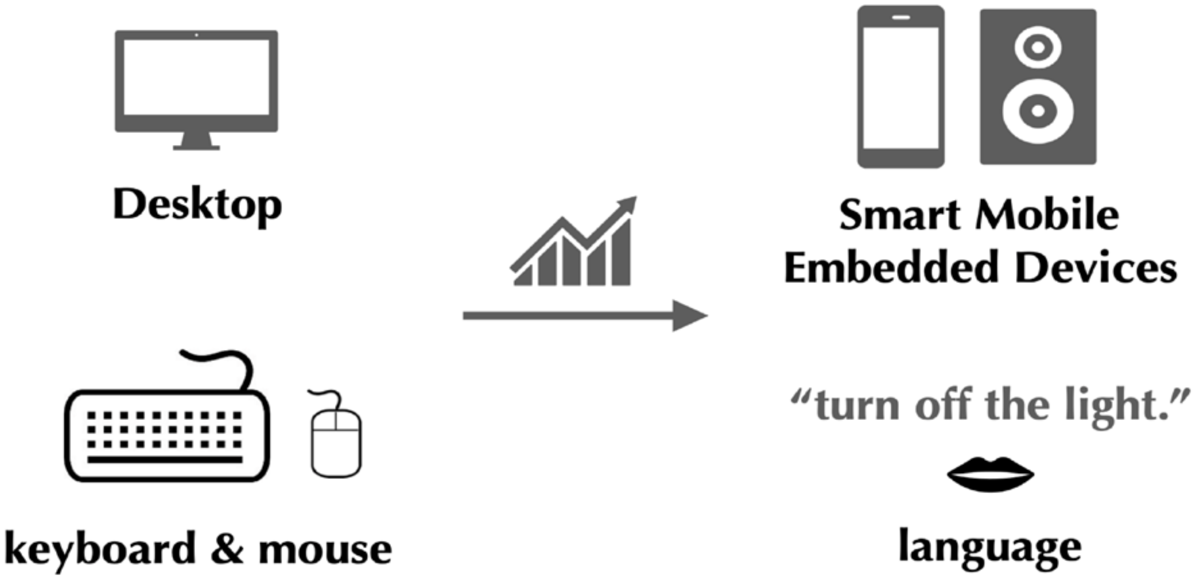
Conversational Agents

Dialogue Systems and Chatbots

Conversational Agents aka Dialogue Systems



- Digital Assistants
- Answering questions on websites
- Communicating with robots
- Chatting for fun
- Clinical uses



Two Classes of Dialog Systems

1. **Task-Oriented Dialogue Agents**
   - Goal-Based Agents
   - Siri, interface with robots, booking flights or hotels

2. **Chatbots**
   - Systems designed for extended conversations
   - Chatting for fun and entertainment

Challenging properties of human conversation

- Taking turns during conversation
- Speech acts
- Grounding
- Dialogue structure
- Initiative
- Implicature

Turn taking

- A conversation is a sequence of turns
  - where you take a turn and then I take a turn.
  - A turn can be a sentence, or a single word.
- A system must know when to start and stop talking.
- Spoken dialogue systems must also detect whether a user is done speaking, so they can process the utterance and respond. This task of **endpoint detection** is tricky because people often pause mid-turn.

Speech acts

Constatives: committing the speaker to something's being the case (*answering*, *claiming*, *confirming*, *denying*, *disagreeing*, *stating*)
Directives: attempts by the speaker to get the addressee to do something (*advising*, *asking*, *forbidding*, *inviting*, *ordering*, *requesting*)
Commissives: committing the speaker to some future course of action (*promising, planning, vowing, betting, opposing*)
Acknowledgments: express the speaker's attitude regarding the hearer with respect to some social action (*apologizing*, *greeting*, *thanking*, *accepting an acknowledgment*)

Conversational Implicature

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

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

- Notice that the client does not in fact answer the agent's question. The speaker expects the hearer to draw certain inferences. **Implicature** means certain kinds of inferences.

Part I: Chatbots

- Systems designed for extended conversations.
- Chatbots mimic unstructured conversations or ‘chats’ that are characteristic of informal human-human interaction
- Architecture include: Rule-Based
  - Pattern-action rules (Eliza)
- Corpus-Based
  - Information Retrieval
  - Neural network encoder-decoder

ELIZA: Weizenbaum (1966)

Simulates a psychologist

Rogerian psychology is a rare type of conversation where “assume the pose of knowing almost nothing of the real world”

Draw the patient out by reflecting patient’s statements back at them

ELIZA: Weizenbaum (1966)

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.

ELIZA Pattern Transform Rules

- Pattern matching rules:
- .* YOU **(.*)** ME **–>** WHAT MAKES YOU THINK I **$1** YOU?

- This rule would transform
  - You **must really hate** me
- into
  - WHAT MAKES YOU THINK I **MUST REALLY HATE** YOU?

Simplified sketch of ELIZA

- **function** ELIZA GENERATOR(user *sentence*) **returns** *response*
- Find the word *w* in *sentence* that has the highest keyword rank
  - **if** *w* exists
    - Choose the highest ranked rule *r* for *w* that matches *sentence*
      response Apply the transform in *r* to *sentence*
    - **if** *w* = 'my'
      - *future* Apply a transformation from the 'memory' rule list to *sentence*
      - Push *future* onto memory stack
  - **else** (no keyword applies)
    - **either**
      - *response* Apply the transform for the NONE keyword to *sentence*
    - **or**
      - *response* Pop the top response from the memory stack
- **return**(*response*)

Modern Chatbots

Two Main Architectures

1. Information Retrieval
2. Machine Learned Sequence Transduction

- Focus on generating a single response turn that is appropriate given the user's immediately previous utterance or two

Conversational Data

- Need: large collections of human conversations

Conversational threads on Twitter or Weibo Retrieve dialog

from movies, indexing subtitles

- Recorded telephone conversations, collected for speech research

Crowdsourced conversations via Mechanical Turk and ParlAI

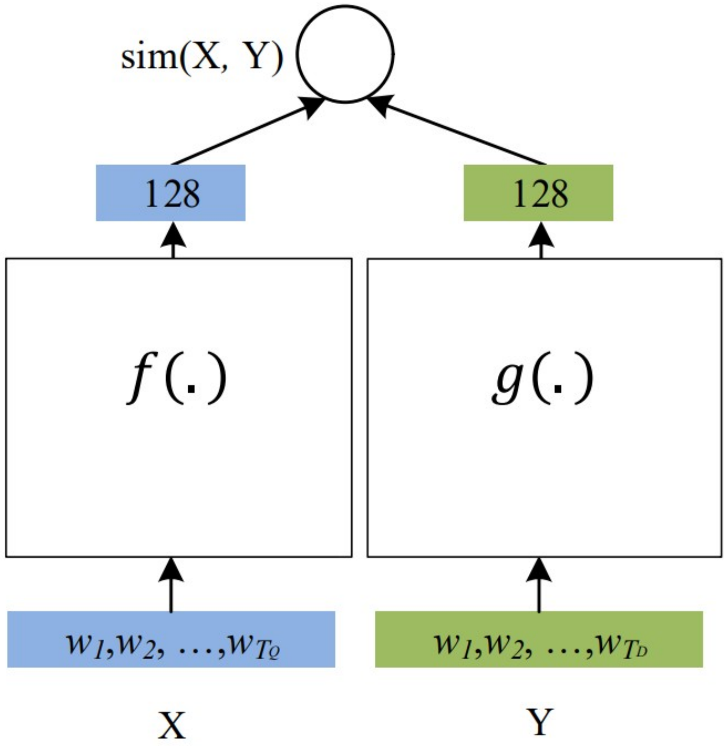
Information Retrieval based Chatbots

- Treat the human user’s input as a query vector **q**
- Search over a large corpus **C** of conversation to find the closest matching turn **t’** in those previous conversations.
- Return the response **r** to that conversational turn. $t' = \arg\max_{t \in C}$ cosine_similarity(q, t).
- r = response(t’)

- q = Have you watched Doctor Who?
- t’ = Do you like Doctor Who?
- r = Yes, I love SciFi shows

IR with Neural Network-Based Similarity Model

Relevance measured by cosine similarity



Word sequence x_t

Learning: maximize the similarity between X (source) and Y (target)

Representation: use DNN to extract abstract semantic features, f or g is a

- Multi-Layer Perceptron (MLP) if text is a bag of words [Huang+ 13]
- **Convolutional Neural Network (CNN) if text is a bag of chunks** [Shen+ 14]
- Recurrent Neural Network (RNN) if text is a sequence of words [Palangi+ 16]

IR-based Models

- Can use more features than just words in query q
  - User features - Information about the user or sentiment
  - Prior turns – Use conversation so far

Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day

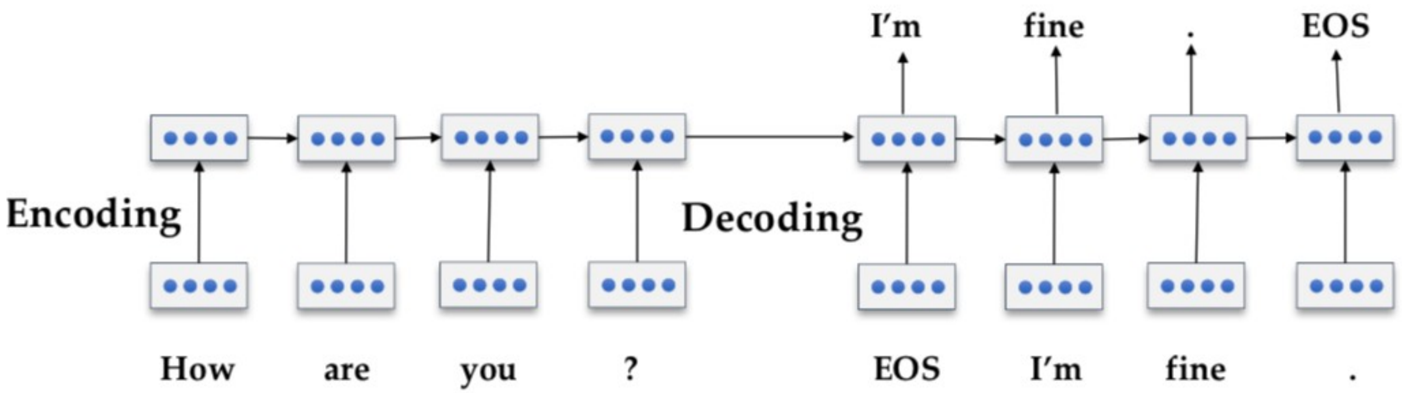
It took less than 24 hours for Twitter to corrupt an innocent AI chatbot. Yesterday, Microsoft [unveiled Tay] — a Twitter bot that the company described as an experiment in "conversational understanding." The more you chat with Tay, said Microsoft, the smarter it gets, learning to engage people through "casual and playful conversation."

Unfortunately, the conversations didn't stay playful for long. Pretty soon after Tay launched, people starting tweeting the bot with all sorts of misogynistic, racist, and Donald Trumpist remarks. And Tay — being essentially a robot parrot with an internet connection — started repeating these sentiments back to users, proving correct that old programming adage: flaming garbage pile in, flaming garbage pile out.

https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist

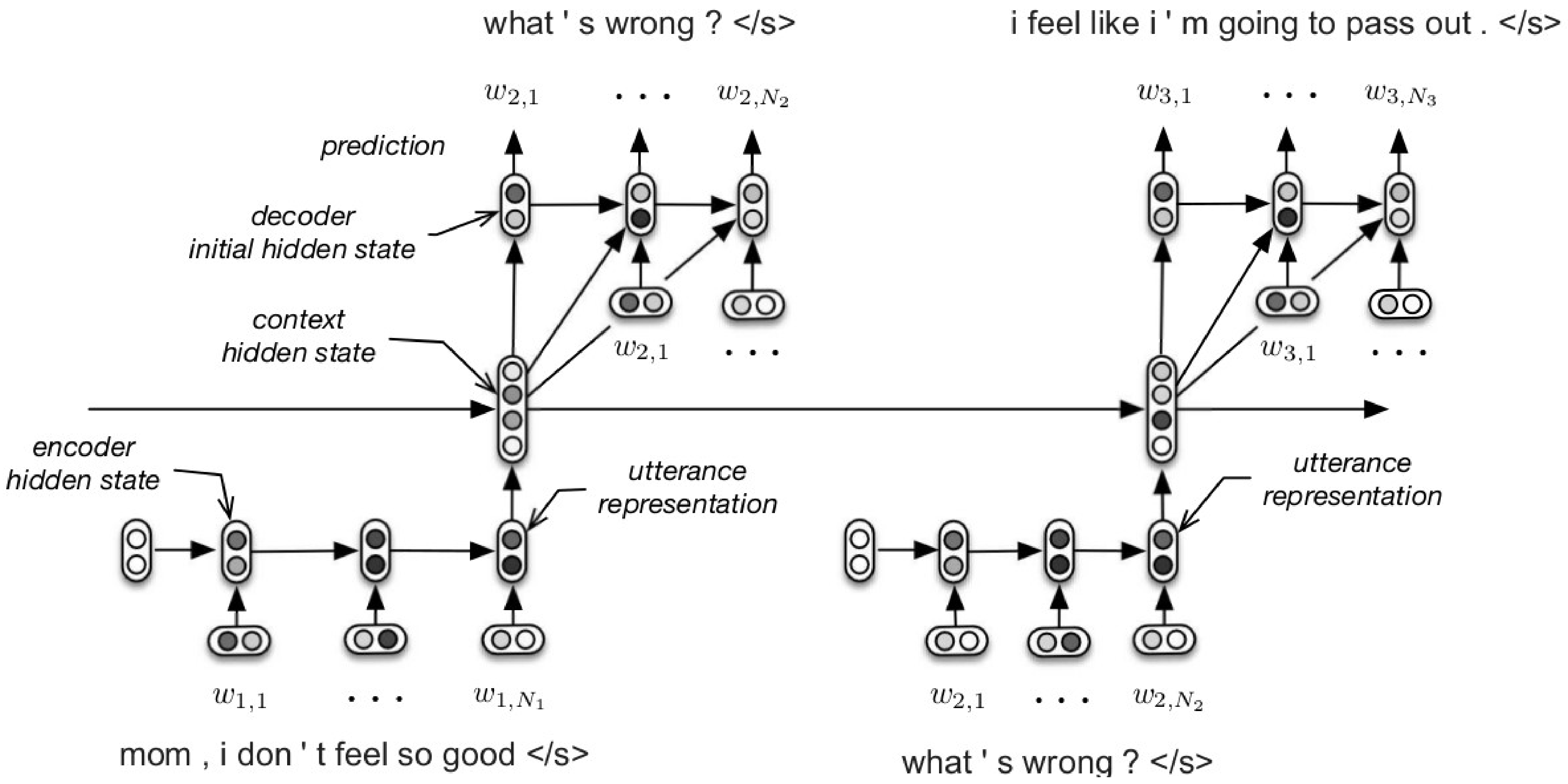
Neural Chatbots

- Think of response generation as a task of transducing from the user's prior turn to the system's turn
- Resp



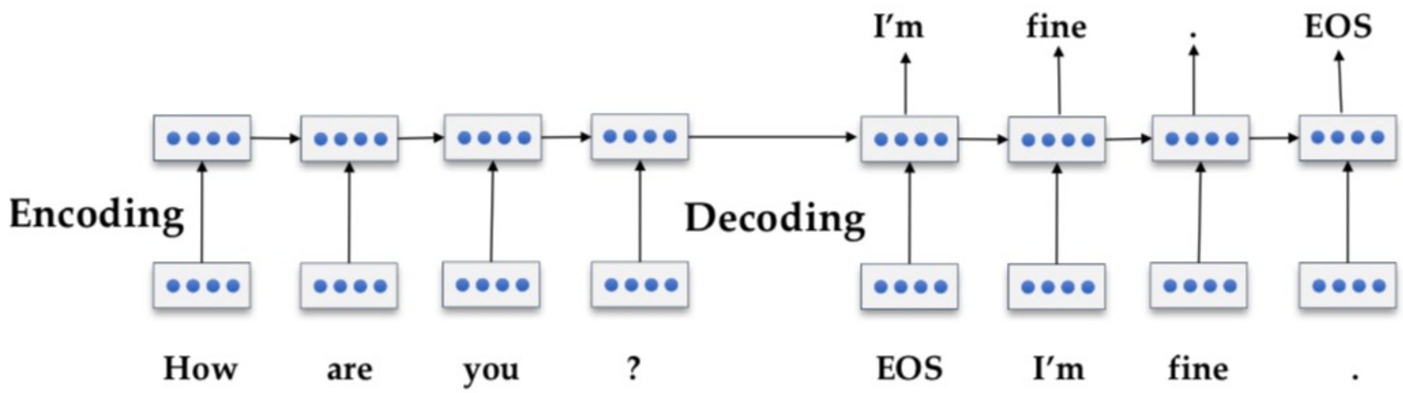
  - Train a deep neural network
    - Map from user1 turn to user2 response

Seq2seq Architecture



Neural Chatbots

- Think of response generation as a task of transducing from the user's prior turn to the system's turn
- Response generation using encoder-decoder models



- Train a deep neural network
  - Map from user1 turn to user2 response

Frame-based Dialogue Systems

- Task-based Dialogue Agents
- Based on “Domain Ontology”
  - A set of “Frames”
- Frame:
  - A knowledge structure representing user intentions
  - A collection of “slots”
  - Each ”slot” having a set of “values”

Example: Travel Domain

Slot : Origin City

Type : City

Value : San Francisco

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?

GUS system : An actual dialogue

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

Slot types can be complex

The type *DATE*

```
DATE
  MONTH:NAME   YEAR:INTEGER    DAY:(BOUNDED-INTEGER 1 31)
  WEEKDAY:(MEMBER (Sunday Monday Tuesday Wednesday
                  Thursday Friday Saturday))
```

Control structure for frame- based dialog

- 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

Natural language understanding for filling slots in GUS

1. Domain classification

- Asking weather? Booking a flight? Programming alarm clock?

2. Intent Determination

- Find a Movie, Show Flight, Remove Calendar Appt

3. Slot Filling

- Extract the actual slots and fillers

Natural language understanding for filling slots in GUS

```
Show me morning flights from Boston to
SF    Tuesday.
on
```

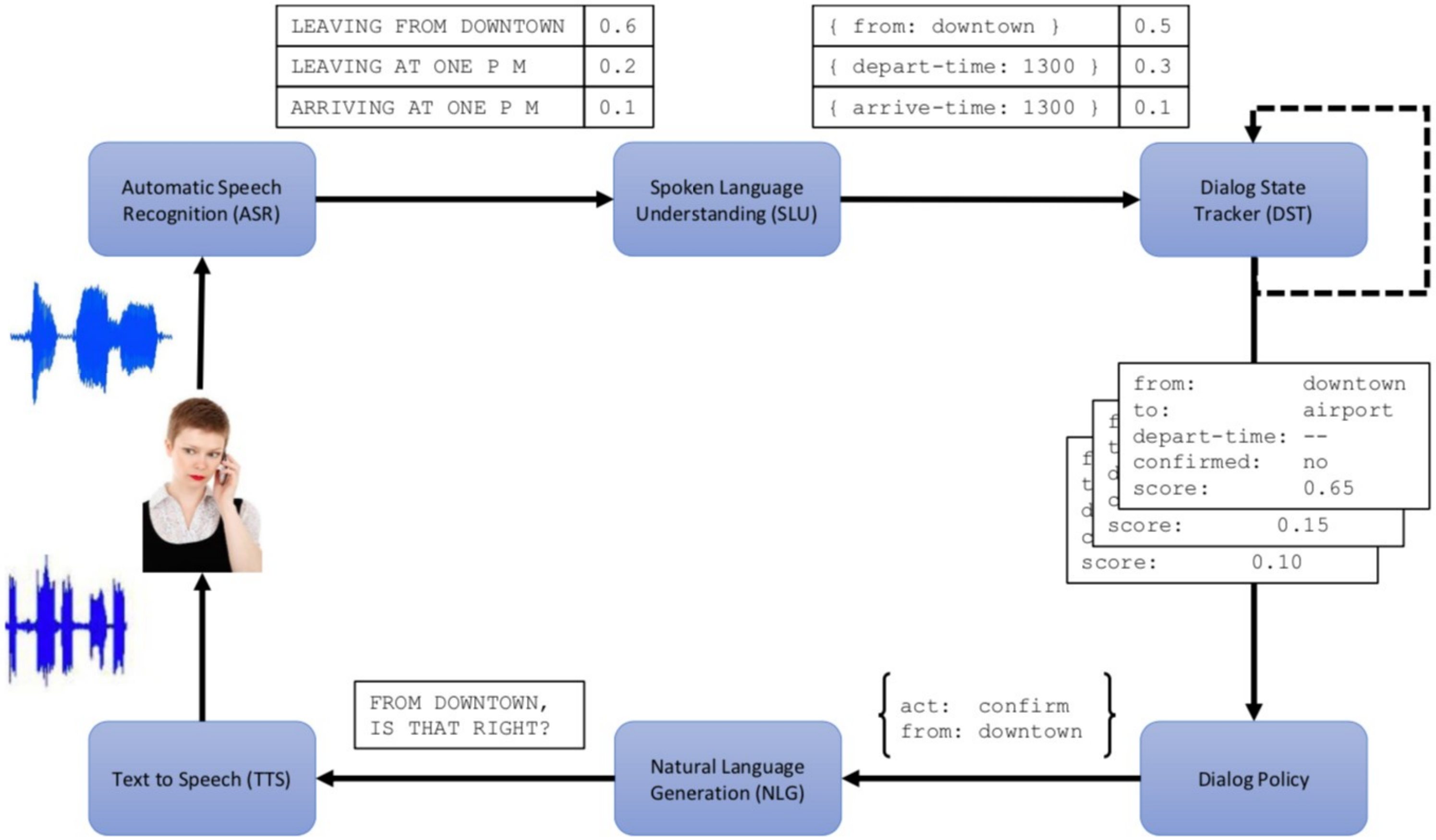
```
DOMAIN:       AIR-TRAVEL
INTENT:       SHOW-FLIGHTS
ORIGIN-CITY:  Boston
ORIGIN-DATE:  Tuesday
ORIGIN-TIME:  morning
DEST-CITY:    San Francisco
```

Rule Sets

- Collections of **rules** consisting of:
  - condition
  - action
- When user input is processed, facts added to store and
  - rule conditions are evaluated
  - relevant actions executed

Dialogue-State Architecture

- More sophisticated version of frame-based architecture



Williams et al. (2016)

- NLU Component:
  - Extract slot fillers using Machine Learning rather than rules
- Dialogue State Tracker:
  - Maintains current state of dialogue, user’s most recent dialogue act
- Dialogue policy:
  - Decides what the system should do or say next
  - When to answer user’s questions, when to make a suggestion
- Natural Language Generation Component:
  - Condition on exact context to produce turns that seem much more natural

Dialogue Acts

- Combining idea of speech acts and grounding into a single representation

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

Dialogue acts used by a restaurant recommendation system (Young et al. (2010))

Dialogue Acts

Sample dialogue from the Recommender System of Young et al. (2010)

	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

- Supervised semantic parsing
- Model to map from input words to slot fillers, domain and intent
- Given a set of labeled sentences
  - "I want to fly to San Francisco on Tuesday"
  - Destination: SF Depart-date: Tuesday
- Requirements: Lots of labeled data

Slot Filling

- *“I want to fly to San Francisco on Monday afternoon please”*

- Use 1-of-N classifier (Naive Bayes, Logistic Regression, Neural Network, etc.)
  - Input:
    - features like word N-grams
  - Output:
    - Domain: AIRLINE Intent: SHOWFLIGHT

More sophisticated algorithm for Slot Filling: IO Tagging

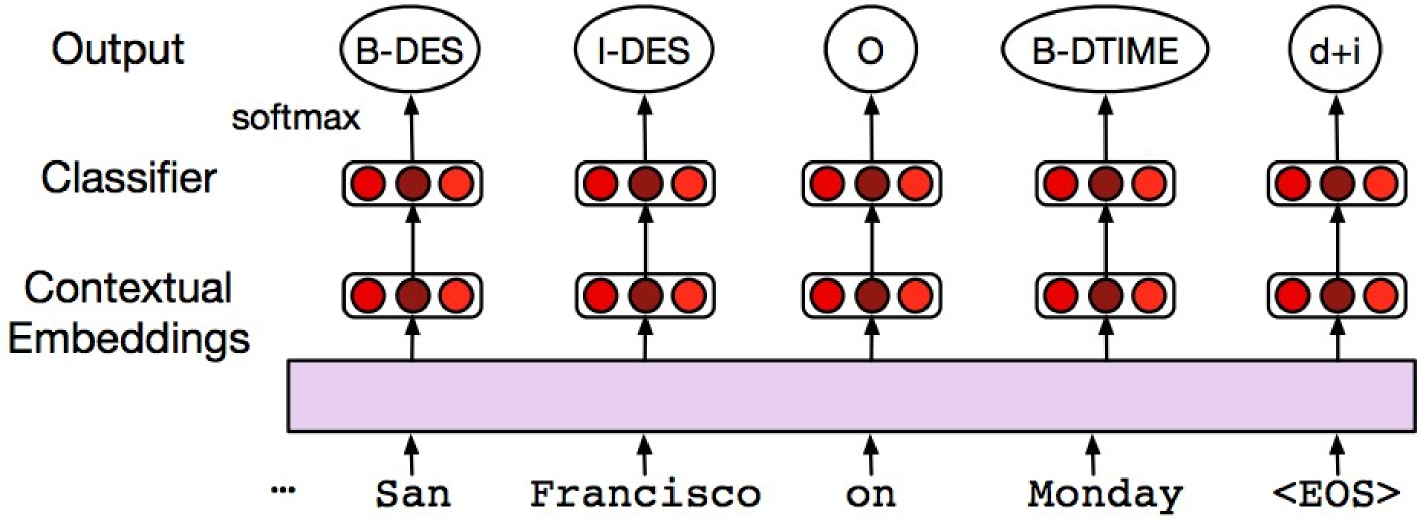
- IOB Tagging
  - Tag for the beginning (B) and inside (I) of each slot label,
  - plus one for tokens outside (O) any slot label
  - 2*n* + 1 tags, where *n* is the number of slots

```
0 0    0   0    0   B-DES  I-DES      0   B-DEPTIME  I-DEPTIME    0
I want to  fly  to  San    Francisco  on  Monday     afternoon    please
```

B-DESTINASTION
I-DESTINATION
B-DEPART_TIME
I-DEPART_TIME
O

Training Data: Sentences paired with sequences of IOB labels

Slot Filling



Simple Architecture for slot filling, mapping the words in the input through contextual embeddings to an output classifier layer

Dialogue State Tracker

- Keep track of
  - Current state of the frame (the fillers of each slot)
  - User’s most recent dialogue act

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

Sample output of a dialogue state tracker after each turn

Dialogue Policy

- What action the system should take next
- What dialogue act to generate
- Predict which action A_i to take

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

A = Dialogue Acts from System; U = Dialogue Acts from User

- Simplification: Condition just on the current dialogue state

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}}\, P(Ai|(Frame_{i-1}, A_{i-1}, U_{i-1})$$

Policy Example: Confirmation and Rejection

- Explicit Confirmation

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

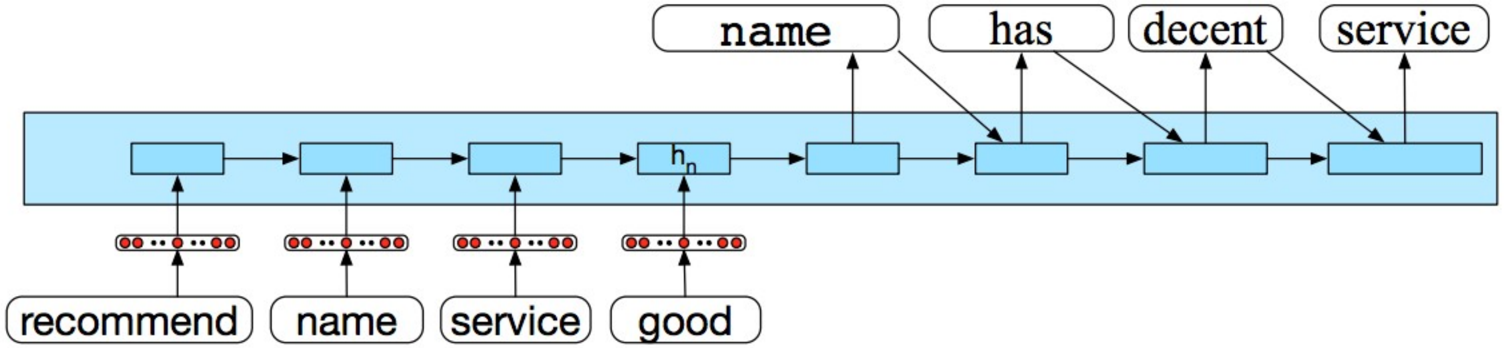
U2: Hi I'd like to fly to Seattle Tuesday Morning
A3: **Traveling to Seattle on Tuesday, August eleventh in the morning.** Your full name?

Natural Language Generation

Modeled in two stages:

- Content Planning (what to say)
- Sentence Realization (how to say it)

Encoder Decoder Models : Map frames to sentences



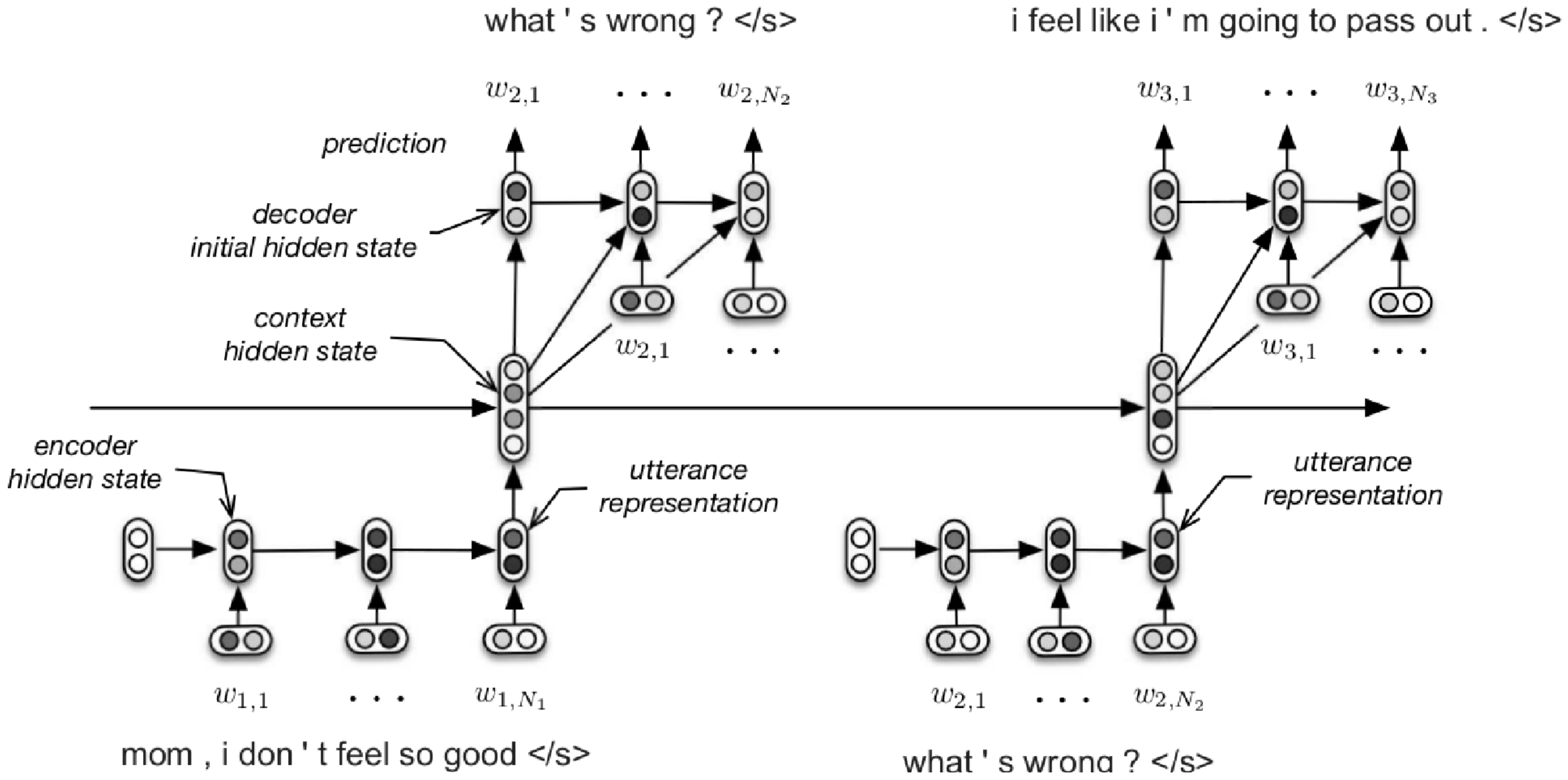
An encoder decoder sentence realizer mapping slots/fillers to English

Neural chatbots

- Think of response generation as a task of *transducing* from the user's prior turn to the system's turn.
- Train on:
  - movie dialogue databases
  - Twitter conversations
- Train a deep neural network
  - map from user1 turn to user2 response

Seq2seq architecture

Serban, Iulian V., Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.



Sample neural chatbot output

Input	tell me ... how long have you had this falling sickness ?
Vanilla-SEQ2SEQ	i'm not a doctor.
Adversarial	a few months, i guess .
Input	so i had the doctors test sammy 's response to conditioning .
Vanilla-SEQ2SEQ	sammy wrote the test sammy wrote the test .
Adversarial	so he took the pills .
Input	they didn 't have much success with this problem commander .
Vanilla-SEQ2SEQ	they 're not the only ones who have been in the system .
Adversarial	can we find someone else ?

?

Ohh I've never seen that! How long does it take you guys to learn the drill?

Like 2 weeks ago!!

Chatbots: pro and con

- Pro:
  - Fun
  - Applications to counseling
  - Good for narrow, scriptable applications
- Cons:
  - They don't really understand
  - Rule-based chatbots are expensive and brittle
  - IR-based chatbots can only mirror training data
    - The case of Microsoft Tay
      - (or, Garbage-in, Garbage-out)
- The future: combining chatbots with frame-based agents