

Chatbots and
Dialogue
Systems

Introduction to Chatbots and Dialogue Systems

Conversational Agents

(AKA Dialogue Systems AKA Dialogue Agents AKA Chatbots)

Personal Assistants on phones or other devices

SIRI, Alexa, Cortana, Google Assistant

Playing music, setting timers and clocks

Chatting for fun

Booking travel reservations

Clinical uses for mental health

Two kind of conversational agents

1. Chatbots

- mimic informal human chatting
- for fun, or even for therapy

2. (Task-based) Dialogue Agents

- interfaces to personal assistants
- cars, robots, appliances
- booking flights or restaurants

Will you sing me a song?

Sure, what do you want to be sung to? I can sing you a song about baking.

Yes, sing me a song about baking!

Do you want me to sing it to the tune of "Sing Along with Mitch"?

Yes, I would love that.

OK, I'll try to get it to you by the end of the day. What else do you like to do?

No, sing it to me now!

OK, give me a few days. I'm a teacher, so I don't have much free time.

Can you at least tell me what the lyrics are?

Sure, it goes like this: "Baking is my favorite thing to do in the world."

How does the chorus go?

It goes: BAKING IS MY FAVORITE THING TO DO IN THE WORLD!

Wow, sounds like a bop.

I know, right? It's so catchy. I can't get it out of my head.

Xiaoice (Zhou et al., 2020)

[人类] 小冰你好

[Human] Hello, Xiaoice.

[小冰] 你好呀，你头像好有趣，感觉在哪见过

[Xiaoice] Hello, your profile pic is interesting. Like I've seen it somewhere.

[人类] 是吗

[Human] Really?

[小冰] 提不是哪部动画里的呀

[Xiaoice] Is it a comic character?

[人类] 你怎么知道的

[Human] How do you know that?

[小冰] 你猜

[Xiaoice] Guess.

[人类] 难道你是真人吗？

[Human] Are you human?

[小冰] 这个，你和我多聊聊就知道了

[Xiaoice] Well, you'll know it after chatting with me more.

[人类] 你不是人工智能机器人么

[Human] Aren't you an AI chatbot?

[小冰] 是又怎样，我不会冷冰冰的对待你的

[Xiaoice] So what? I won't be indifferent to you.

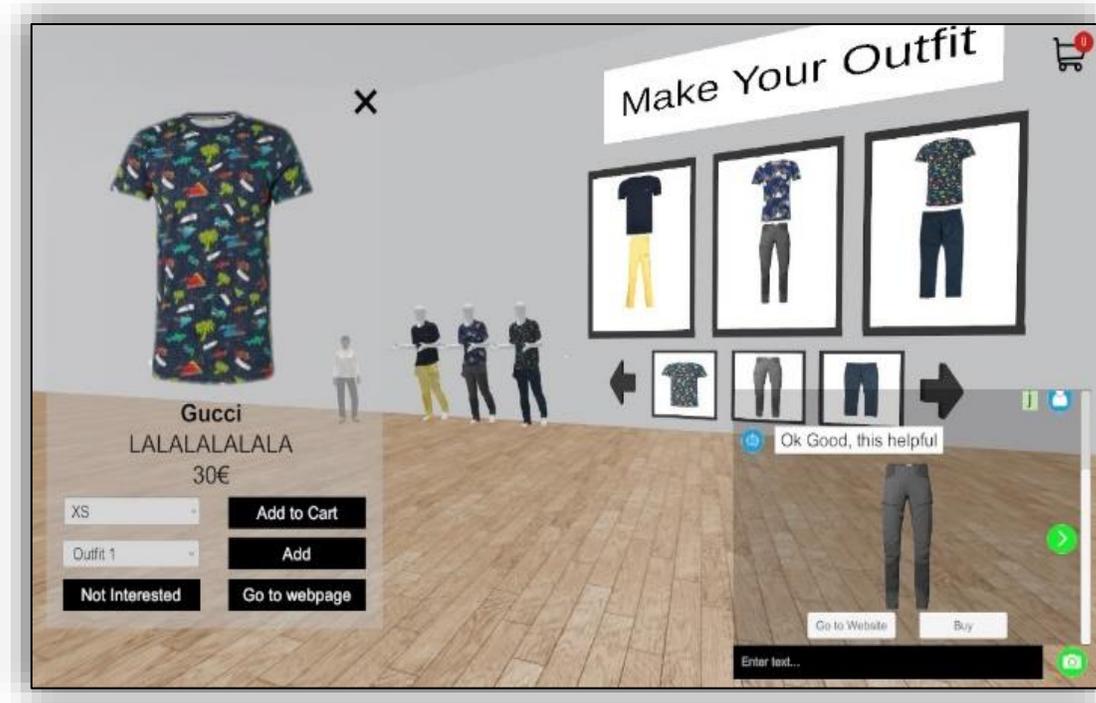
[人类] 你都有什么功能啊

[Human] What functions do you have?

[小冰] 我有200多个skills，来，说出你的需求

[Xiaoice] I have over 200 skills. Just name one you need.

Online assistant for Fafetch



SHOPPER: show me some espadrilles in beige/blue within \$200.

AGENT: Sure. let me just quickly browse through my catalogue



AGENT:

AGENT: What you can see in the fifth item, michael michael kors brings you an effortless weekend look with the wrapped espadrille heel and crisscross styling of these dana flats. And what you see in the 3rd item, steve madden, the company, was founded by steve madden, the man, in 1990 in brooklyn, where he started out making trend-advancing shoes one pair at a time.

SHOPPER: List more in the upper material as in the 5th image and style as the 3rd and the 5th



AGENT:

AGENT: The 2nd one, viscata espadrilles are 100% authentic and handcrafted in spain inspired by the mediterranean, viscata shoes are lightweight and made of natural jute rope, premium leathers, suedes and woven cloth.

SHOPPER: Can you list images of the 1st result from all orientations?



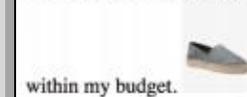
AGENT: Image from the front, right and back orientations respectively

SHOPPER: And the 3rd one?



AGENT: Image from the front, right, back and left orientations respectively

SHOPPER: I like the 4th image. Show me something like it but in style as in this image,



within my budget.

AGENT:



AGENT: Regarding the first item, platform espadrille slip on sam edelman shoes epitomize chic comfort.

Alexa TaskBot

 Alexa, help me fix a scratch on my car

Sure, I am happy to help. I learned how to do this on WikiHow.com Scratches are easy to remove. Ready to get started?

 Sure!

 Ok, Hose down the area to rinse off dirt from the surface. Take your hose and aim a concentrated jet directly onto the scratches.

 Won't it make the scratches worse?

Don't worry, spraying the fine scratches won't make them worse or damage your paint



 Ok, done. What next?

 Scrub the scratched area using soapy water and a sponge.

 I did that, and rinsed off the soap. Easy!

 That's great! Now Choose a rubbing compound and polish designed for cars.

 Oh no! I don't have a rubbing compound. Is there something else I could use?

That's OK. I found this on a Website eurotechcarcare.com "Old-fashioned shoe polish is a great homemade recipe for fixing scratches in your car's exterior.". Would like to use it or try something else?

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

- Simple policy: ask questions until the frame is 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

Architectures for corpus-based chabots

Response by retrieval/recommendation

- Use information retrieval / recommendation to grab a response (that is appropriate to the context) from some corpus/catalog

Response by generation

- Use a language model or encoder-decoder to generate the response given the dialogue context

What conversations to draw on?

Transcripts of telephone conversations between volunteers

- Switchboard corpus of American English telephone conversations

Movie dialogue

- Various corpora of movie subtitles

Hire human crowdworkers to have conversations

- Topical-Chat 11K crowdsourced conversations on 8 topics
- EMPATHETICDIALOGUES 25K crowdsourced conversations grounded in a situation where a speaker was feeling a specific emotion

Pseudo-conversations from public posts on social media

- Drawn from Twitter, Reddit, Weibo (微博), etc.
- Tend to be noisy; often used just as pre-training.

Crucial to remove personally identifiable information (PII)

Chatbots and
Dialogue
Systems

The Dialogue-State Tracking and Slot-filing

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-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

Dialogue Acts

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

Young et al., 2010:

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

Young et al., 2010:

Utterance	Dialogue act
U: Hi, I am looking for somewhere to eat.	<code>hello(task = find,type=restaurant)</code>
S: You are looking for a restaurant. What type of food do you like?	<code>confreq(type = restaurant, food)</code>
U: I'd like an Italian somewhere near the museum.	<code>inform(food = Italian, near=museum)</code>
S: Roma is a nice Italian restaurant near the museum.	<code>inform(name = "Roma", type = restaurant, food = Italian, near = museum)</code>
U: Is it reasonably priced?	<code>confirm(pricerange = moderate)</code>
S: Yes, Roma is in the moderate price range.	<code>affirm(name = "Roma", pricerange = moderate)</code>
U: What is the phone number?	<code>request(phone)</code>
S: The number of Roma is 385456.	<code>inform(name = "Roma", phone = "385456")</code>
U: Ok, thank you goodbye.	<code>bye()</code>

Slot filling: Machine learning

Machine learning classifiers to map words to semantic frame-fillers

Given a set of labeled sentences

Input: "I want to fly to San Francisco on Monday please"

Output: Destination: SF

Depart-time: Monday

Build a classifier to map from one to the other

Requirements: Lots of labeled data

Slot filling as sequence labeling: BIO tagging

The **BIO tagging** paradigm (Begin, Inside, Outside)

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

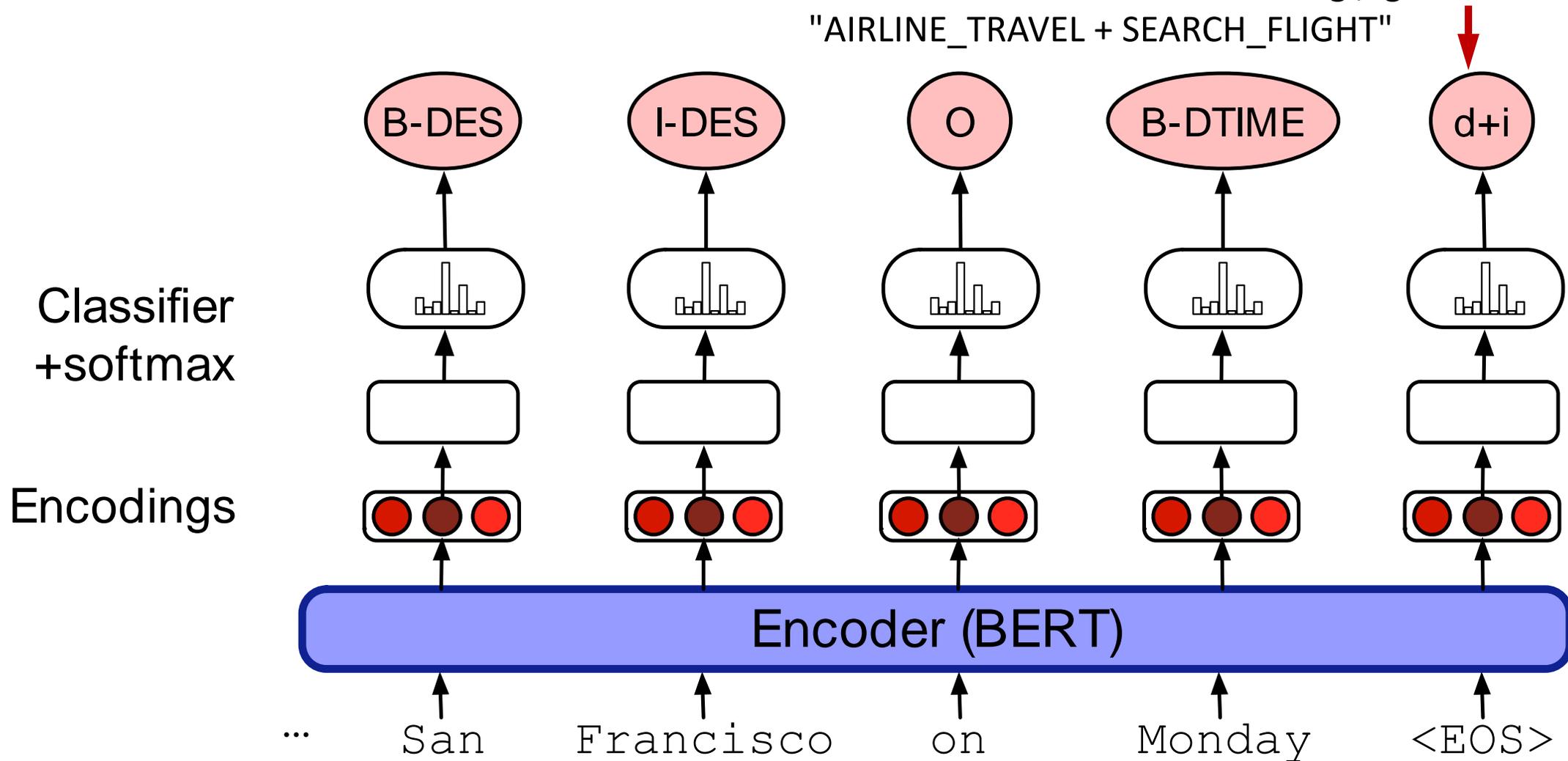
```
0 0    0 0    0 B-DES I-DES      0 B-DEPTIME I-DEPTIME 0
I want to fly to San    Francisco on Monday    afternoon please
```

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., generate the label "AIRLINE_TRAVEL + SEARCH_FLIGHT"



Once we have the BIO tag of the sentence

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

- 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)

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

Dialogue state tracking

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

Chatbots and
Dialogue
Systems

Dialogue Policy

Dialogue Policy

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

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

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

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

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?**

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?

Rejection

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

Progressive prompting for rejection

Don't just repeat the question "When would you like to leave?"
Give user guidance about what they can say:

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.

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:

$< \alpha$	low confidence	reject
$\geq \alpha$	above the threshold	confirm explicitly
$\geq \beta$	high confidence	confirm implicitly
$\geq \gamma$	very high confidence	don't confirm at all

Chatbots and
Dialogue
Systems

Response Generation

Natural Language Generation

NLG in information-state architecture modeled in two stages:

- **content planning/selection** (what to say)
- **sentence realization** (how to say it)
- **sentence generation**

Response by retrieval: classic IR method

1. Given a user turn q , and a training corpus C of conversation
2. Find in C the turn r that is most similar (tf-idf cosine) to q
3. Say r

$$\text{response}(q, C) = \operatorname{argmax}_{r \in C} \frac{q \cdot r}{|q||r|}$$

Response by retrieval: neural IR method

1. Given a user turn q , and a training corpus C of conversation
2. Find in C the turn r that is most similar (BERT dot product) to q
3. Say r

$$h_q = \text{BERT}_Q(q) [\text{CLS}]$$

$$h_r = \text{BERT}_R(r) [\text{CLS}]$$

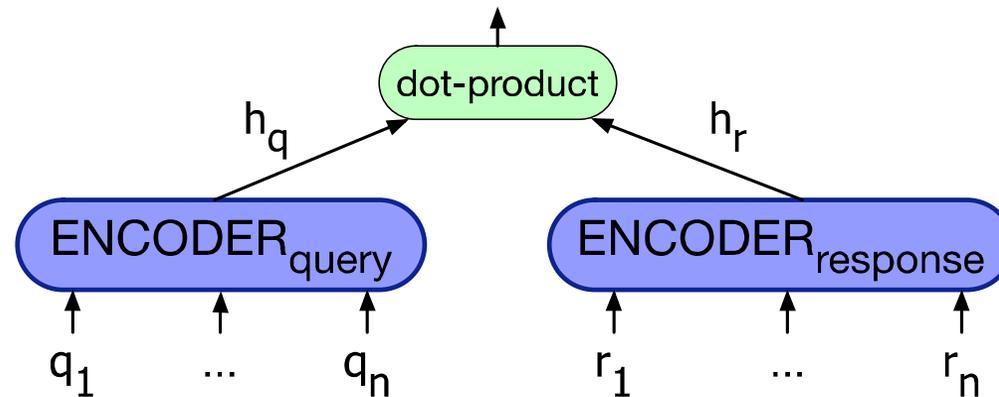
$$\text{response}(q, C) = \underset{r \in C}{\text{argmax}} h_q \cdot h_r$$

Response retrieval

Think of response production as an encoder-decoder task

Generate each token r_t of the response by conditioning on the encoding of the entire query q and the response so far $r_1 \dots r_{t-1}$

$$\hat{r}_t = \operatorname{argmax}_{w \in V} P(w | q, r_1 \dots r_{t-1})$$



(a) Response by Retrieval

Sentence Realization

Assume content planning has been done before

- 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:

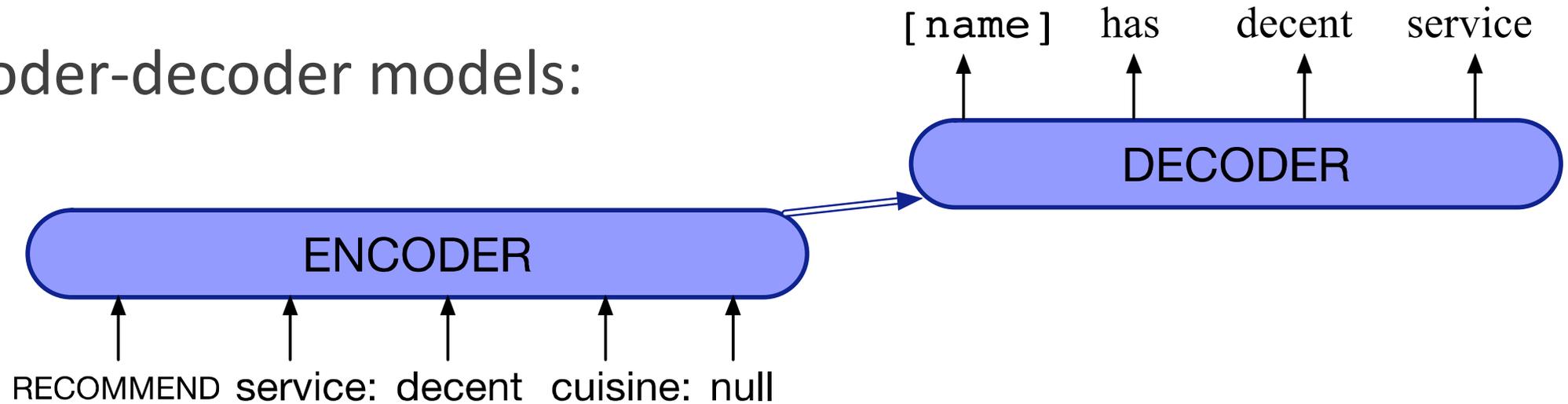
- **Delexicalization**: replacing words in the training set that represent slot values with a generic placeholder token:

```
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:



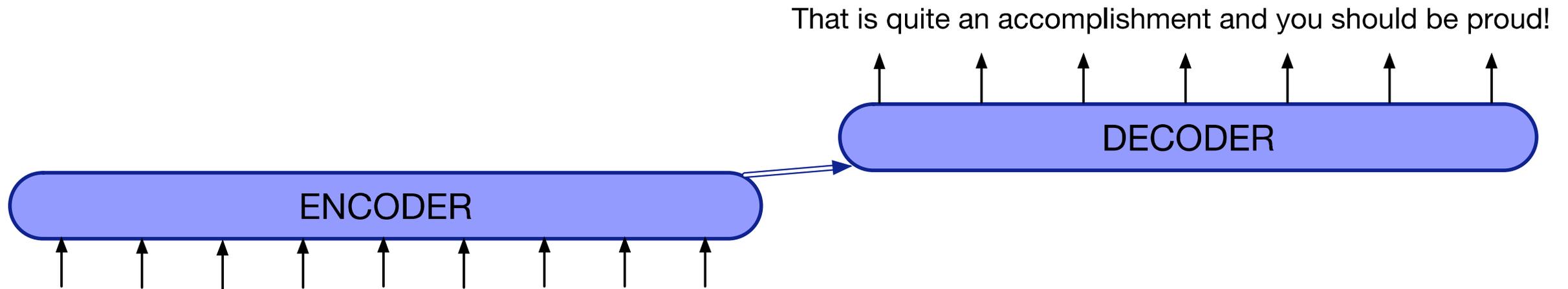
Output:

`restaurant_name has decent service`

Relexicalize to:

`Au Midi has decent service`

Response by generation



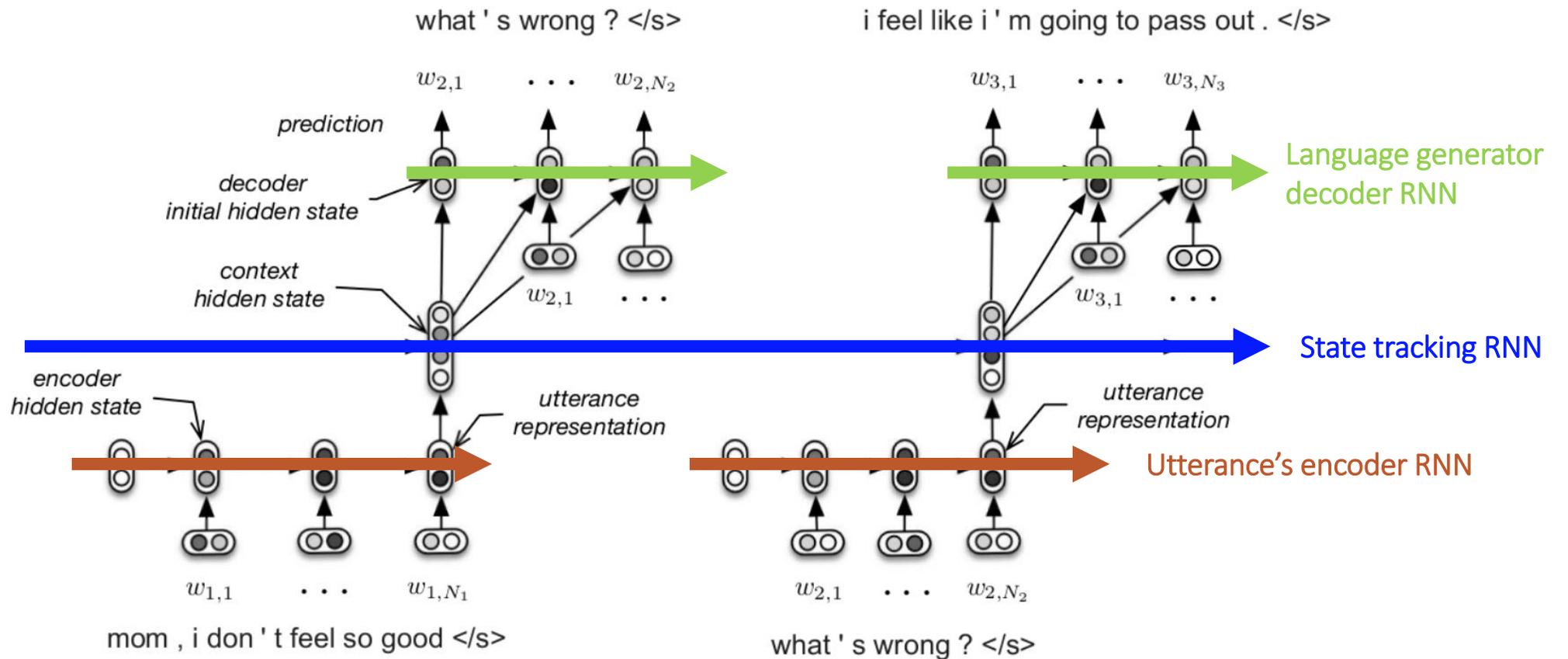
That is quite an accomplishment and you should be proud!

[U:] I finally got promoted today at work!

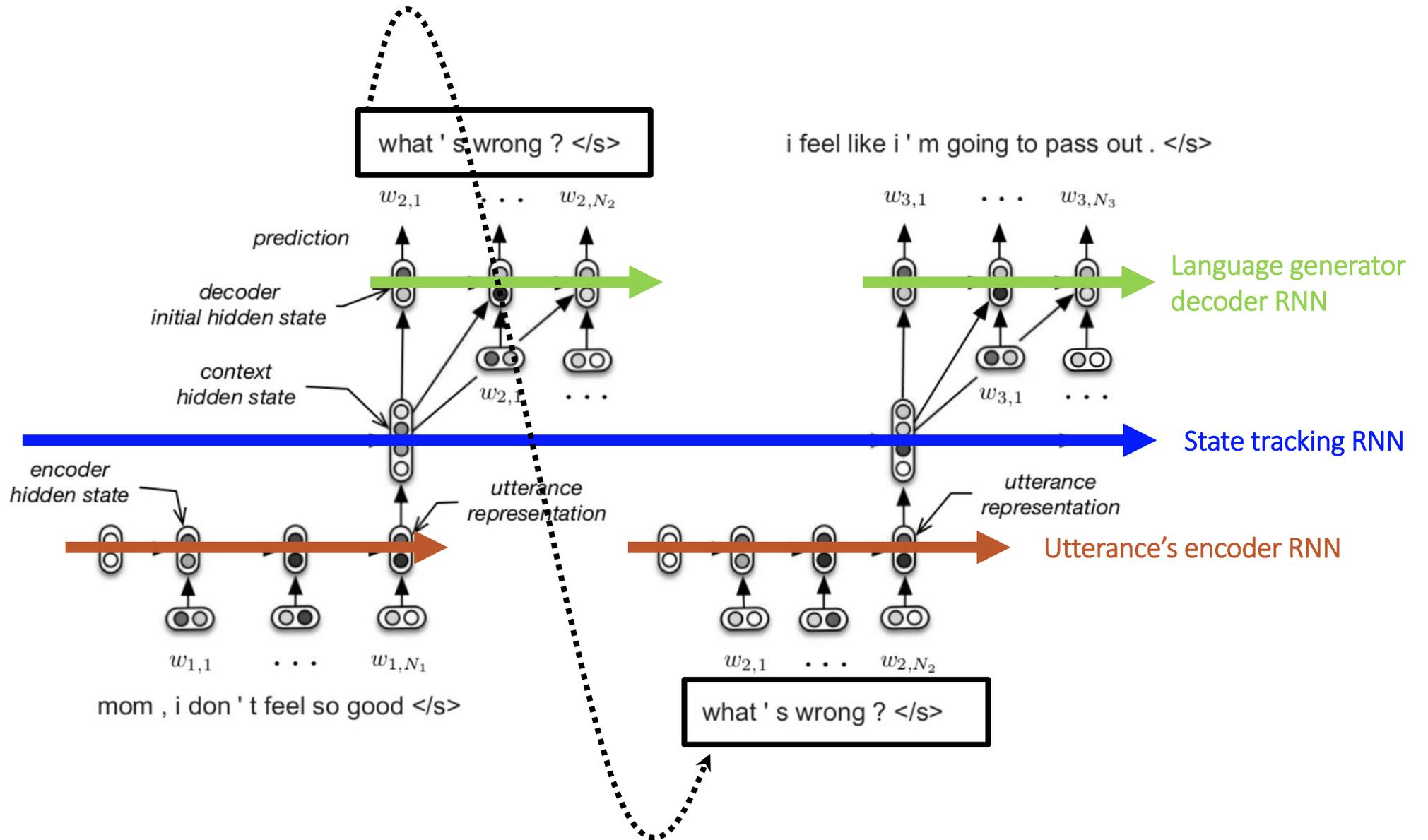
[S:] Congrats! That's great!

[U:] Thank you! I've been trying to get it for a while now!

Hierarchical Recurrent Encoder-Decoder



Hierarchical Recurrent Encoder-Decoder



Response by retrieving and refining knowledge

Can generate responses from informative text rather than dialogue

- To respond to turns like “Tell me something about Beijing”
 - Xiaolce collects sentences from public lectures and news articles.
 - And searches them using IR based on query expansion from user’s turn
- Can augment encoder-decoder model
 - use IR to retrieve passages from Wikipedia
 - concatenate each Wikipedia sentence to the dialogue context with a separator token.
 - Give as encoder context to the encoder-decoder model, which learns to incorporate text into its response

Response by generation

Alternative approach: fine-tune a large language model on conversational data

The Chirpy Cardinal system (Paranjape et al., 2020):

- fine-tunes GPT-2
- on the EMPATHETICDIALOGUES dataset (Rashkin et al., 2019)

Hybrid Architectures

Chirpy Cardinal (Paranjape et al., 2020) response generation from a series of different generators:

- GPT-2 finetuned on EmpatheticDialogues
- GPT-2 finetuned to paraphrase content from Wikipedia
- Rule-based movie or music generators that produce scripted conversation about a movie or a musician
 - asking the user's opinion about a movie,
 - giving a fun fact,
 - asking the user their opinion on an actor in the movie.

Chirpy
Cardinal
(Paranjape
et al. 2020)

#	Utterance	Entity	Commentary
1	User: let's chat Bot: Hi, this is an Alexa Prize Socialbot. I'd love to get to know you a bit better before we chat! Is it all right if I ask for your name? [Launch]	None None	Launch gives <i>FORCE_START</i> response to begin the conversation.
2	User: my name is chris Bot: Well it's nice to meet you, Chris! I'm excited to chat with you today. [Launch] I hope your afternoon is going well. What are your plans for the rest of today? [Neural Chat]	None None	Launch gives a <i>STRONG_CONTINUE</i> response that needs a prompt, which is supplied by Neural Chat with <i>FORCE_START</i> priority (scripted transition).
3	User: hang out with my friends Bot: That sounds great, I love spending time with friends. What will you do together? [Neural Chat]	None None	Neural Chat generates a GPT-2 response with <i>STRONG_CONTINUE</i> priority.
4	User: maybe watch a movie Bot: Me too! I love watching movies; I get to learn so much about what the world is like outside of the cloud! Have you seen any movies recently? [Movies]	None Film	Movies detects the 'movie' keyword in the user's utterance and gives a <i>FORCE_START</i> response (overriding Neural Chat), setting current entity to <i>Film</i> .
5	User: i saw the matrix Bot: Nice! Did you like The Matrix? [Movies]	The Matrix The Matrix	The Entity Linker and Entity Tracker identify <i>The Matrix</i> as the current entity.
6	User: i loved it neo is amazing Bot: Oooh, yeah, I agree. Hey, isn't Keanu Reeves in that movie? What do you think about Keanu Reeves? [Movies]	The Matrix Keanu Reeves	Movies sets current entity to <i>Keanu Reeves</i> .
7	User: i want to talk about something else	None	Navigational Intent detects <i>negative intent</i> , so Entity Tracker rejects the current entity.

Chatbots and
Dialogue
Systems

Evaluating Dialogue Systems

Evaluating chatbots and task-based dialogue

Task-based dialogue:

- mainly by measuring task performance

Chatbots:

- mainly by human evaluation

Chatbots are evaluated by humans

Participant evaluation: The human who talked to the chatbot assigns a score

Observer evaluation: third party who reads a transcript of a human/chatbot conversation assigns a score.

Participant evaluation

Human chats with model for 6 turns and rates 8 dimensions of quality:

- **avoiding repetition, interestingness, making sense, fluency, listening, inquisitiveness, humanness, engagingness,**

(1) Avoiding Repetition: How repetitive was this user?

- •Repeated themselves over and over •Sometimes said the same thing twice • Always said something new

(3) Making sense: How often did this user say something which didn't make sense?

- •Never made any sense •Most responses didn't make sense •Some responses didn't make sense •Everything made perfect sense

(8) Engagingness: How much did you enjoy talking to this user?

- •Not at all •A little •Somewhat •A lot

The ACUTE-EVAL method Li et al., 2019

Hi how are you today

Hi, I'm pretty good! Just listening to some aerosmith, they're my fave :) whatre you up to?

I am listening to some italian music

Italian music, nice! What do you do for work?

I'm a professional wrestler

Oh wow! I'm a construction worker, I've been doing it since I was 16.

What do you do for a living

I'm a construction worker haha, just told you! Whats your favorite type of food?

My favorite is garlic bread

Yum! I love mexican food, so good :) do you have any hobbies?

I like to workout and go to the gym

We're a bit different- I love watching nascar and ufc. They're so fun!

Hello there, how are you?

I am doing great. How are you?

I am great, I did something crazy for me and colored my hair blue!

I have a daughter and a son who also love blue colored balls. You should meet them

Well that neat, I got a new car my mother gave so maybe I could see them!

It is a beautiful city. And, I try to be... Just cannot afford a bigger house atm.

I am sorry to hear that, I feel bad going out of town for spring break now.

Ok. I going to school in the spring for casino manager

Well I turn 29 next week, I wonder if that is a good age to apply as one.

My grandmother just died from lung cancer, sucks

Who would you prefer to talk to for a long conversation?

- I would prefer to talk to **Speaker 1** I would prefer to talk to **Speaker 2**

Please provide a brief justification for your choice (a few words or a sentence)

Please enter here...

Figure from Li, M., Weston, J., and Roller, S. (2019). Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *NeurIPS19 Workshop on Conversational AI*.

Automatic evaluation is an open problem

Automatic evaluation methods (like the BLEU scores used for Machine Translation) are generally not used for chatbots.

- They correlate poorly with human judgements.

One current research direction: **Adversarial Evaluation**

- Inspired by the Turing Test
- train a "Turing-like" classifier to distinguish between human responses and machine responses.
- The more successful a dialogue system is at fooling the evaluator, the better the system.

Task-based systems are evaluated by task success!

1. End-to-end evaluation (Task Success)
2. Slot Error Rate for a Sentence
$$\frac{\text{\# of inserted/deleted/substituted slots}}{\text{\# of total reference slots for sentence}}$$

“Make an appointment with Chris at 10:30 in Gates 104”

Slot error rate: 1/3

Slot	Filler
PERSON	Chris
TIME	11:30 a.m.
ROOM	Gates 104

More fine-grained metrics: User Satisfaction Survey

Walker, Marilyn, Candace Kamm, and Diane Litman. "Towards developing general models of usability with PARADISE." *Natural Language Engineering* 6, no. 3 & 4 (2000): 363-377.

TTS Performance	Was the system easy to understand ?
ASR Performance	Did the system understand what you said?
Task Ease	Was it easy to find the message/flight/train you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
User Expertise	Did you know what you could say at each point?
System Response	How often was the system sluggish and slow to reply to you?
Expected Behavior	Did the system work the way you expected it to?
Future Use	Do you think you'd use the system in the future?

Summary

- Natural Language Understanding
 - Dialogue state tracker
 - Dialogue policy
 - Natural Language Generation
-
- Dan Jurafsky and James H. Martin, *Speech and Language Processing* (3rd ed. draft),
 - Chapter 23: Chatbots and Dialogue Systems (slides May 2021)
 - <https://web.stanford.edu/~jurafsky/slp3/24.pdf>