CS388: Natural Language Processing

Lecture 13: Instruction Tuning, RLHF, Dialog

Greg Durrett

TEXAS

The University of Texas at Austin

Announcements

- Project 3 tips:
  - We highly recommend using a GPU (including Colab)
  - You don't need all training iterations
  - You can decrease the frequency of checkpointing
- Project 2 back soon
- Final project proposals back soon

Recap: Chain-of-thought

Prompt

Input	**Context**: *Christopher agrees with Kevin.* [...] **Q**: *Who hangs out with a student?*
Label+ Explanation	***Mary***, ***because Mary hangs out with Danielle and Danielle is a student.***
Train Ex	
Train Ex	
Test Input	**Context**: *Adam plays with Ellen.* [...] **Q**: *Who plays with a doctor?*

GPT-3

Output	*Adam*, ***because Adam plays with Ellen and Ellen is a doctor.***
	greedy decoding from GPT-3

Recap: Chain-of-thought

- Can help substantially on mathematical reasoning
- Some work to optimize the specifics of the prompts and the examples

Input: Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

...

Q: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
A:

Model output: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. So that is 10 x .5 = 5 hours a day. 5 hours a day x 7 days a week = 35 hours a week. The answer is 35 hours a week. ✓

Wei et al. (2022)

Today

- Instruction tuning
- RLHF/DPO
- Chatbots
- Task-oriented dialogue systems

Instruction Tuning
(= Supervised Fine-Tuning (SFT))

Instruction Tuning

- We want to optimize models for P(answer | prompt, input), but they're learned on a basic language modeling objective
- One solution: treat the basic language modeling as pre-training, then fine-tune them on what we care about
- Two versions of this:
  - **Instruction tuning:** supervised fine-tuning on data derived from many NLP tasks
  - **Reinforcement learning from human feedback (RLHF):** RL to improve human judgments of how good the outputs are

Types of Data to Learn From

- **Supervised data:** used in instruction tuning (= supervised fine-tuning)
  - Input **x**: *who was the US president during World War II?*
  - Gold output **y***: *Franklin D. Roosevelt, Harry Truman*
- **Preferences:** used in RLHF
  - Input **x**: *who was the US president during World War II?*
  - Outputs $\mathbf{y}^+$: *Franklin D. Roosevelt, Harry Truman*

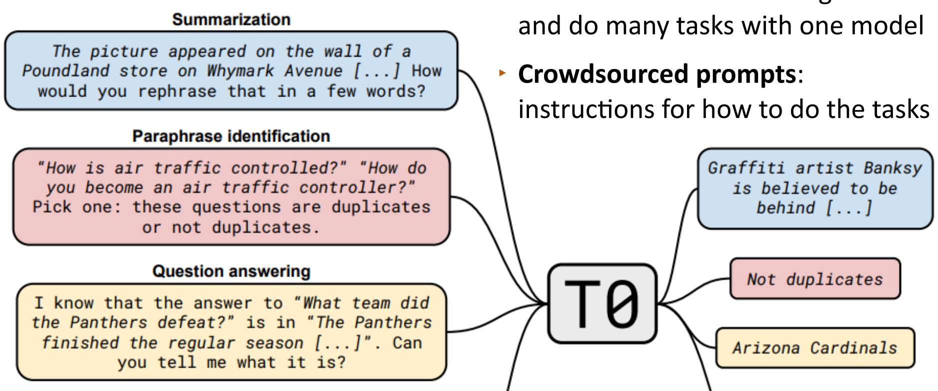
    $\mathbf{y}^-$: *Herbert Hoover, Franklin D. Roosevelt, Harry Truman*

    $\mathbf{y}^+$: *Franklin D. Roosevelt until April 12, 1945, then Harry Truman after Roosevelt died*

    $\mathbf{y}^-$: *Franklin D. Roosevelt, Harry Truman*

Task Generalization: T0

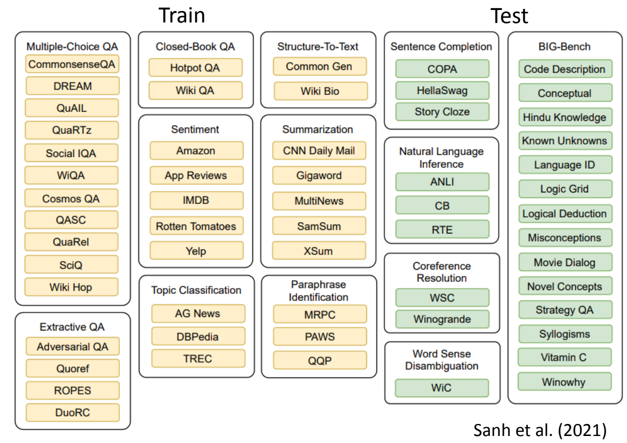
- T0: tries to deliver on the goal of T5 and do many tasks with one model
- **Crowdsourced prompts**: instructions for how to do the tasks



Sanh et al. (2021)

Task Generalization

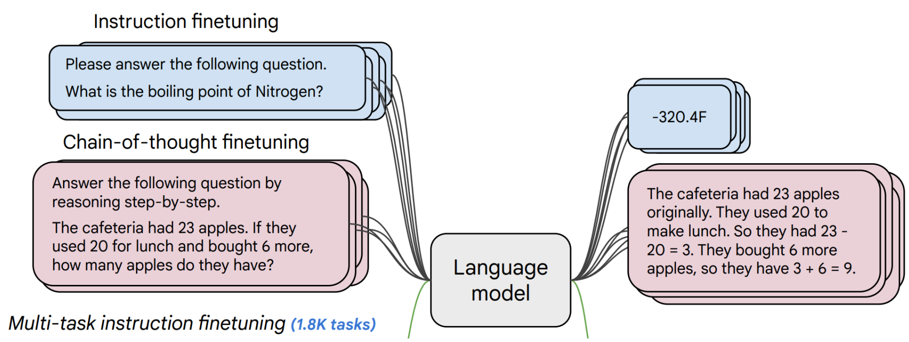
- Pre-train: T5 task
- Train: a collection of tasks with prompts. **This uses existing labeled training data**
- Test: a new task specified only by a new prompt. **No training data in this task**



Sanh et al. (2021)

Flan-PaLM

- Flan-PaLM (October 20, 2022): 1800 tasks, 540B parameter model fine-tuned on many tasks after pre-training



Chung et al. (2022)

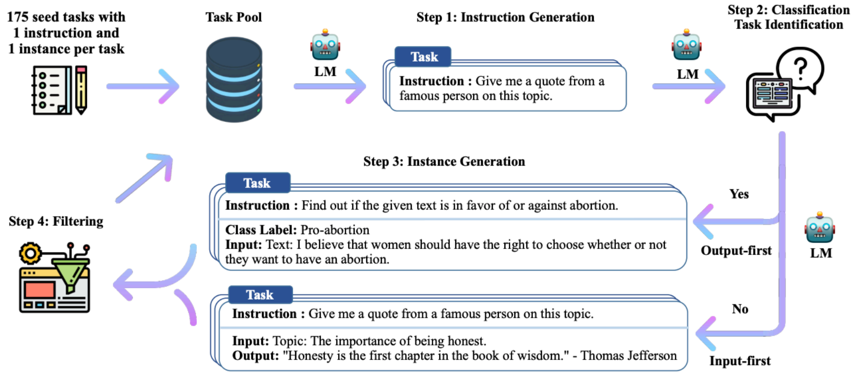
Flan-PaLM

				MMLU		BBH	
Model	Finetuning Mixtures	Tasks	Norm. avg.	Direct	CoT	Direct	CoT
540B	None (no finetuning)	0	49.1	71.3	62.9	49.1	63.7
	CoT	9	52.6 (+3.5)	68.8	64.8	50.5	61.1
	CoT, Muffin	89	57.0 (+7.9)	71.8	66.7	56.7	64.0
	CoT, Muffin, T0-SF	282	57.5 (+8.4)	72.9	68.2	57.3	64.0
	CoT, Muffin, T0-SF, NIV2	1,836	58.5 (+9.4)	73.2	68.1	58.8	65.6

- Human performance estimates are ~80 on Big-Bench (BBH)

Chung et al. (2022)

Self-Instruct/Alpaca



- Fine-tune Llama on 52k outputs with answers generated by text-davinci-003

Yizhong Wang et al. (2023) Self-Instruct

Ronen Taori et al. (2023) Alpaca

LIMA

Source	#Examples	Avg Input Len.	Avg Output Len.
Training			
Stack Exchange (STEM)	200	117	523
Stack Exchange (Other)	200	119	530
wikiHow	200	12	1,811
Pushshift r/WritingPrompts	150	34	274
Natural Instructions	50	236	92
Paper Authors (Group A)	200	40	334
Dev			
Paper Authors (Group A)	50	36	N/A
Test			
Pushshift r/AskReddit	70	30	N/A
Paper Authors (Group B)	230	31	N/A

Table 1: Sources of training prompts (inputs) and responses (outputs), and test prompts. The total amount of training data is roughly 750,000 tokens, split over exactly 1,000 sequences.

- How little data can we get away with for fine-tuning?

Chunting Zhou et al. (2023)

LIMA

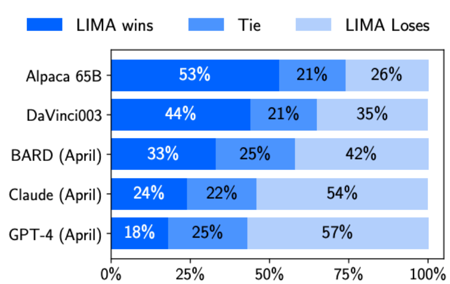


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

Chunting Zhou et al. (2023)

Open Questions

- How much does instruction tuning actually change?
  - "The unlocking spell on base LLMs": analysis showing that alignment only changes the presence of a few tokens, you can get many of the benefits from prompting a base model

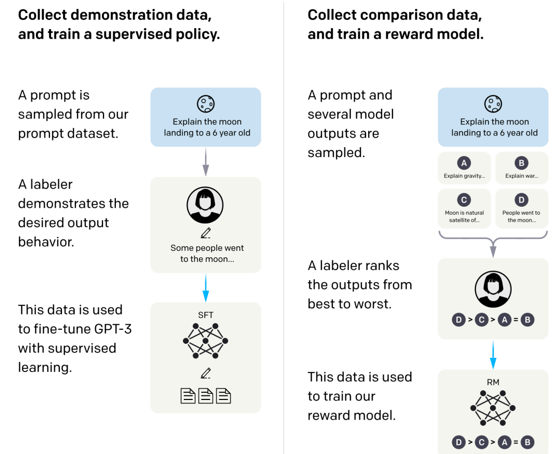
Yuchen Lin et al. (2023)

- Limitations of instruction tuning? "False promises of imitating proprietary LLMs"
  - Suppose you're fine-tuning a 7B-parameter model to imitate a 100B one. You have a QA example where the 100B model gave the right answer. What should the 7B model do?

Gudibande et al. (2023)

Reinforcement Learning from Human Feedback (RLHF)

RLHF



- Apply this approach to optimizing outputs from large language models
- Step 3 (not shown): do RL with this policy

Ouyang et al. (2022)

Learning Reward Models

- Input **x**: *who was the US president during World War II?*
- Outputs **y**⁺: *Franklin D. Roosevelt, Harry Truman*

  y⁻: *Herbert Hoover, Franklin D. Roosevelt, Harry Truman*

$$P(y^+ \succ y^- \mid \mathbf{x}) = \frac{\exp(r(y^+, \mathbf{x}))}{\exp(r(y^+, \mathbf{x})) + \exp(r(y^-, \mathbf{x}))}$$

- This turns scores into log probabilities of 1 being preferred to 2. Same as logistic regression where we classify pairs as 1 > 2 or 2 < 1, but we actually learn a continuous scoring function, not a classifier
- Outcome: reward model r(y, **x**) returning real-valued scores

Ouyang et al. (2022)

RLHF

- Goal: find a policy π_θ (LM parameters) that optimizes the following:

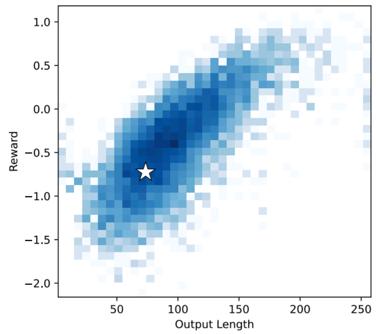
$$R(\mathbf{x}, y) = r(\mathbf{x}, y) - \lambda D_{\mathrm{KL}}(\pi_\theta(y \mid \mathbf{x}) \| \pi_\theta^{\mathrm{SFT}}(y \mid \mathbf{x}))$$

get high reward — stay close to an initial SFT policy

- This is called *proximal policy optimization* (PPO)
- Important to regularize towards the SFT policy! Reward models are not stable enough to make things work

Christiano et al. (2017)

RLHF



Question: *Why don't adults roll off the bed?*

☆ **SFT (Before); 59 tokens**

Adults typically do not roll off of the bed because they have developed the muscle memory to keep their bodies from involuntarily moving during sleep and maintaining proper posture.

RLHF (After); 243 tokens: Similar output, but much longer / more details

Adults generally do not roll off of the bed because they have developed muscle control and awareness which helps them to remain upright while sleeping.

Additionally, most adults find it uncomfortable or even painful to move around in their sleep, so rolling off the bed would be difficult without waking up first. There may also be safety concerns such as falling out of bed and potentially hurting oneself if one were to lose balance or fall from the ...

- Reward models trained on open datasets have high correlations with length

Singhal, Goyal, Xu, Durrett (arXiv 2023)

RLHF

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

- For OpenAI, RLHF data is collected from their API. **Very different from instruct-tuning datasets**

Ouyang et al. (2022)

Direct Preference Optimization (DPO)

- Through some manipulation, it can be shown that the optimal policy π^* for RLHF satisfies the preference model

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

ref = SFT policy. preferred output should be more likely under our learned policy than under reference, dispreferred output should be less likely

- We can now learn the policy directly to optimize the log likelihood of the preference data in a fashion that looks like supervised learning:

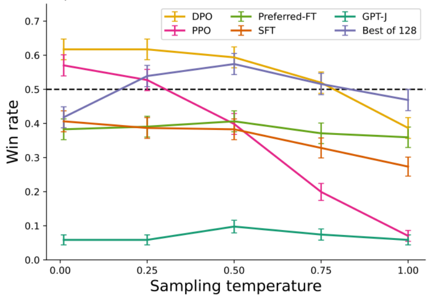
$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

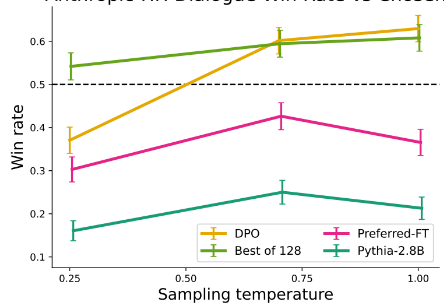
Rafailov et al. (2023)

Outcome of RLHF/DPO

- RLHF produces an "aligned" model that should achieve high reward
- Baselines:
  - Best-of-n: sample n responses from an SFT model, take the best one according to the reward function
    - Pro: training-free
    - Cons: expensive, may not deviate far from the initial SFT model
  - Preference tuning: apply SFT on preferred outputs
    - Pro: simple. Cons: doesn't use the negative examples

Direct Preference Optimization (DPO)





- Evaluation: *win rate* (as scored by an LLM)

Rafailov et al. (2023)

RLHF in practice

Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

RLHF data for Llama 2

- They do 5 iterations of (train, get more preferences, get new reward model). First 3 iterations: just fine-tuning best-of-n, then they used PPO
- Current approaches: many papers exploring versions with active data collection (e.g., tune with DPO -> collect preferences -> keep tuning …)

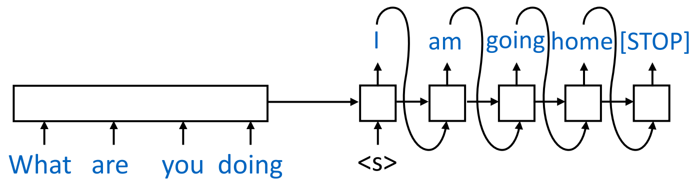
Touvron et al. (2023)

Pre-trained Chatbots

What are chatbots?

- Like story generation in that it's open-ended, but involves dialogue with a user
- Input: a conversation history of utterances, plus something the user (a person) just said.
  Output: the model's response to that
- Needs to generate interesting and diverse content, but also needs to be able to answer questions and carry on a conversation

Seq2seq models



- Can we just train seq2seq models to "translate" from utterance to response?
- Hard to evaluate with automatic metrics:

System	BLEU
RANDOM	0.33
MT	3.21
HUMAN	6.08

Lack of Diversity

- Training to maximize likelihood gives a system that prefers common responses:

Input: What are you doing?	
-0.86 I don't know.	-1.09 Get out of here.
-1.03 I don't know!	-1.09 I'm going home.
-1.06 Nothing.	-1.09 Oh my god!
-1.09 Get out of the way.	-1.10 I'm talking to you.
Input: what is your name?	
-0.91 I don't know.	...
-0.92 I don't know!	-1.55 My name is Robert.
-0.92 I don't know, sir.	-1.58 My name is John.
-0.97 Oh, my god!	-1.59 My name's John.
Input: How old are you?	
-0.79 I don't know.	...
-1.06 I'm fine.	-1.64 Twenty-five.
-1.17 I'm all right.	-1.66 Five.
-1.17 I'm not sure.	-1.71 Eight.

Li et al. (2016)

PersonaChat

Persona 1	**Persona 2**
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Game of Thrones

Zhang et al. (2018)

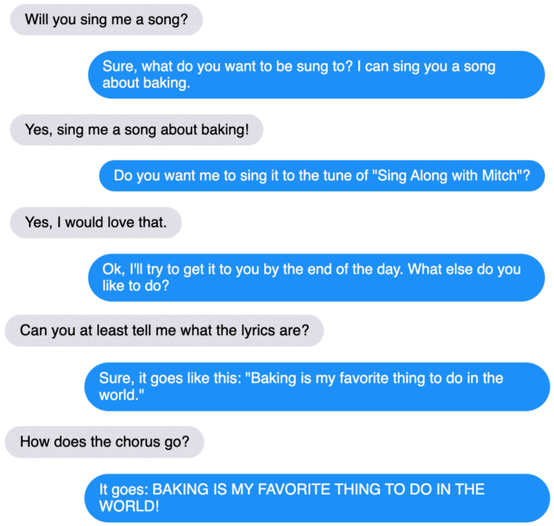
[PERSON 1:] Hi
[PERSON 2:] Hello ! How are you today ?
[PERSON 1:] I am good thank you , how are you.
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thrones.
[PERSON 1:] Nice ! How old are your children?
[PERSON 2:] I have four that range in age from 10 to 21. You?
[PERSON 1:] I do not have children at the moment.
[PERSON 2:] That just means you get to keep all the popcorn for yourself.
[PERSON 1:] And Cheetos at the moment!

- Efforts to imbue seq2seq models with "personality"
- These systems still don't work great. What else is needed?

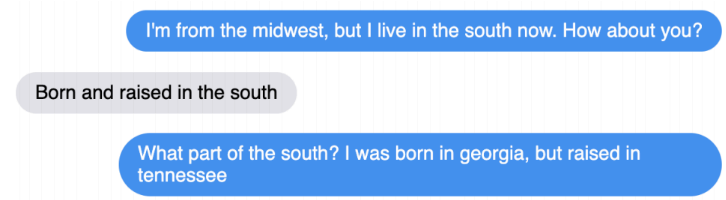
Blender

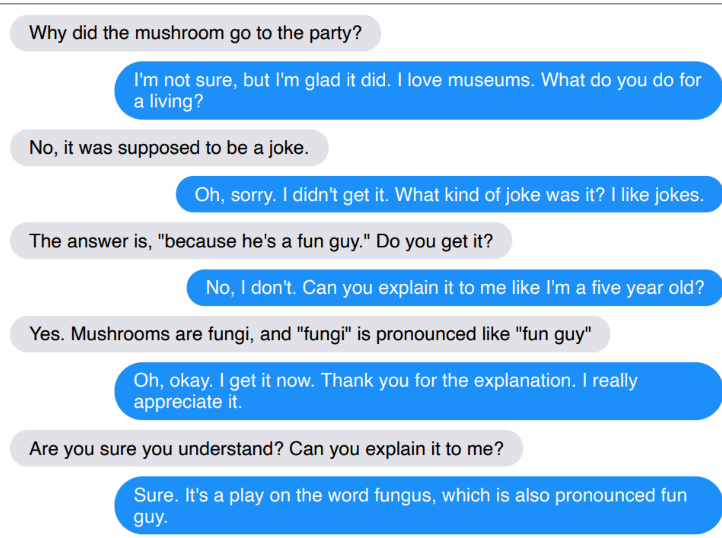
- By 2020: large models + prompting solve many of these problems!
- 2.7B-param model, also a 9.4B-parameter seq2seq model variant
- "Poly-encoder" Transformer architecture, some training tricks
- Three models: retrieve (from training data), generate, retrieve-and-refine
- Fine-tuning on three prior datasets: PersonaChat, Empathetic Dialogues (discuss personal situation, listener is empathetic), Wizard of Wikipedia (discuss something from Wikipedia)

Roller et al. (2020)



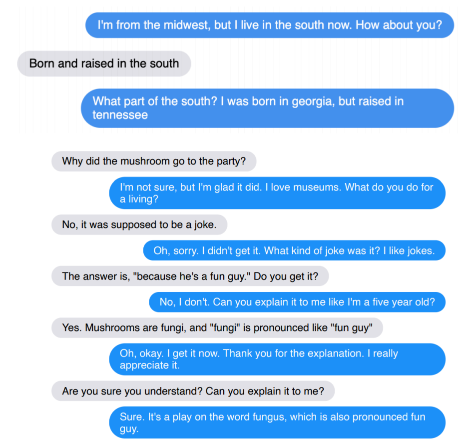
Blender





Blender

- Inconsistent responses: this model doesn't **really** have anything to say about itself
- Holding a conversation != AI
  - Can't acquire new information
  - Did it learn "fun guy"? No, it doesn't understand phonology. It probably had this in the data somewhere



Chatbots

- What happens when these models get really good at fooling people? Google LaMDA model (similar to Blender):


"I KNOW A PERSON WHEN I TALK TO IT" —

Google fires Blake Lemoine, the engineer who claimed AI chatbot is a person

Google says Lemoine violated security rules, slams "wholly un

NEWS

Ex-Google engineer Blake Lemoine discusses sentient AI

Ex-Google engineer Blake Lemoine discusses why LaMDA and other AI systems may be considered sentient and explains exactly how much AI systems know about consumers.

BBC Sign in Home News Sport Reel Worklife Trave

NEWS

Home | War in Ukraine | US Elections 2022 | Coronavirus | Climate | Video | World | US & Canada | UK | Busines

Tech

Blake Lemoine: Google fires engineer who said AI tech has feelings

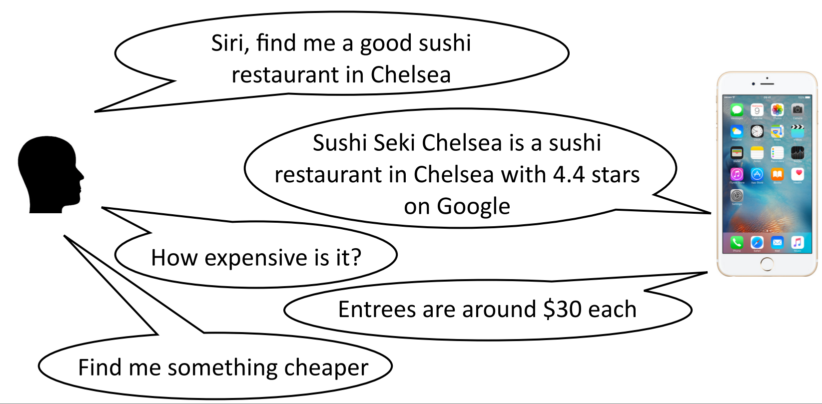
Modern Chatbots

- ChatGPT is not really a chatbot. It's optimized for providing information, not necessarily giving stimulating conversation
- Other services like character.ai are more optimized for conversation
- Alexa Prize chatbots: separate types of models with hand-engineered dialog flows (e.g., if the user mentions a movie, give a piece of trivia about that movie pulled from IMDB)

Task-Oriented Dialogue

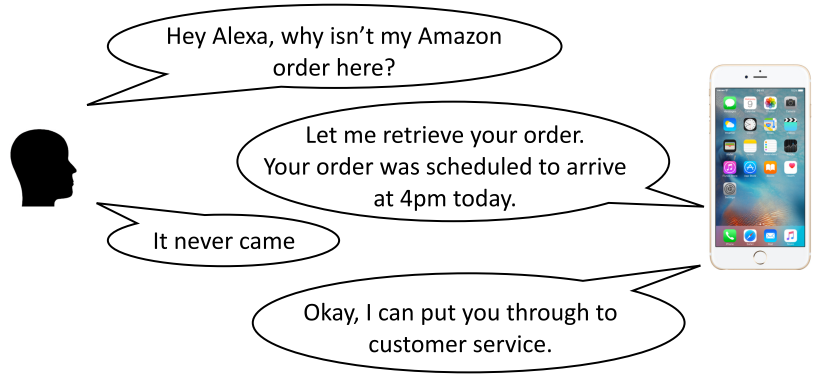
Task-Oriented Dialogue

- How do you build conversational systems to do things?



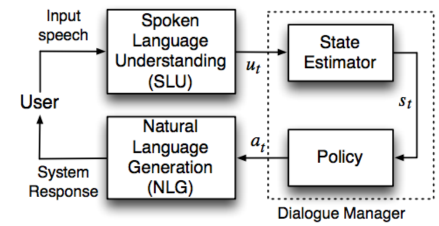
Task-Oriented Dialogue

- Customer service:



Task-Oriented Dialogue

- Parsing / language understanding is just one piece of a system
- Dialogue state: reflects any information about the conversation (e.g., search history)



- User utterance -> update dialogue state -> take action (e.g., query the restaurant database) -> say something
- How do we represent the information from the user's utterance?

Young et al. (2013)

ATIS

- Intent and slots model: classify an intent (**Airfare**), then fill several slots needed to specify the parameters for that intent

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*

- This is how most Alexa skills work. Can match with rule-based systems or use classifiers

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

Intents

- 29 different intents in ATIS:

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*

Dataflow Graphs

- How do we scale to more complex dialog scenarios? One proposal: **dataflow graphs**

(1)

User: *Where is my meeting at 2 this afternoon?*

```
place(findEvent(EventSpec(start=pm(2))))
```

Agent: *It's in Conference Room D.*

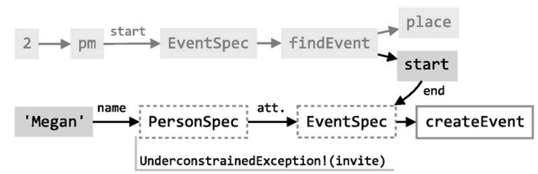
Semantic Machines; Andreas et al. (2020)

Dataflow Graphs

(2)

User: *Can you create a meeting with Megan right before that starts?*

```
createEvent(EventSpec(
  end=start(refer(Constraint[Event]())),
  attendee=PersonSpec(name='Megan')
))
```



Agent: *Which person named Megan did you mean?*

Semantic Machines; Andreas et al. (2020)

Task-Oriented Dialog: What the user sees

Find me a good sushi restaurant in Chelsea

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

How expensive is it?

Entrees are around $30 each

Task-Oriented Dialog: Under the hood

Find me a good sushi restaurant in Chelsea

```
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

Training Dialog Systems

- "Wizard of Oz": can run the dialog system in a real setting and have a human decide what it should do next
- Learning from demonstrations: the system can learn from what the wizard does and do that in the future

Find me a good sushi restaurant in Chelsea

wizard enters these

```
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

Semantic Machines; Andreas et al. (2020)

Task-Oriented Dialogue

- Building these systems takes a ton of engineering, like Gunrock — it typically **doesn't** use pre-trained models (until 2023…)
  - Need to know what the system should **do**, not just what it should say
  - Generation is usually templated (handwritten), otherwise the system can behave unexpectedly
- Lots of industry activity in this space, less in academia (hard to maintain all of the moving parts for a real dialog system)
- Current interest: work like Toolformer / Langchain that allows LLMs to generate the API calls directly

Takeaways

- Instruction-tuning and RLHF/DPO are two procedures that take LMs to the next level — these models work dramatically better than basic LLMs
- These are the foundation of modern chatbots (along with lots of pre-training data), very exciting capabilities in these LLM agents
- Task-oriented dialog has historically been different but is starting to unify with chatbots (Bing agent has ability to make API calls)