

CS769 Advanced NLP

Dialog System

Junjie Hu



Slides adapted from Shikib, Dan
<https://junjiehu.github.io/cs769-spring23/>

Goal for Today

- Brief History of Dialog Systems
- Task-oriented Dialog
- Chatbot
- Evaluation

What is dialog?

- NLP for conversations
 - Understand utterances in the context of the dialog history (i.e., back-and-forth interaction)
 - Generate responses
- a.k.a. conversational agents, chatbot, dialog agents
- Personal assistants on phones or other devices: Siri, Alexa, Cortana, Google Assistant

Two Types of Dialog Systems

- **Open-domain Chatbots**
 - Mimic informal human chatting
 - Chatting for fun, or even for therapy, clinical uses for mental health
 - Question answering
- **Task-oriented Dialog (ToD)**
 - Personal assistants on mobile devices
 - NL interfaces in cars, robots, appliances
 - Customer support, booking flights, hotels, restaurants
- Note: with LLMs, the distinction of these two systems become blurry

Brief History of Dialog Research

Dialog Architectures

- **Rule-based**
 - Pattern-action rules (ELIZA)
 - + A mental model (PARRY): the first system to pass the Turing test!
- **Corpus-based**
 - Information retrieval (Xiaolce)
 - Neural encoder-decoder (BlenderBot)

Real Example: A telephone conversation

- A human travel agent (A) and a human client (C)

C₁: ...I need to travel in May.
A₂: And, what day in May did you want to travel?
C₃: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A₄: And you're flying into what city?
C₅: Seattle.
A₆: And what time would you like to leave Pittsburgh?
C₇: Uh hmm I don't think there's many options for non-stop.
A₈: Right. There's three non-stops today.
C₉: What are they?
A₁₀: The first one departs PGH at 10:00am arrives Seattle at 12:05 their time.
The second flight departs PGH at 5:55pm, arrives Seattle at 8pm. And the
last flight departs PGH at 8:15pm arrives Seattle at 10:28pm.
C₁₁: OK I'll take the 5ish flight on the night before on the 11th.
A₁₂: On the 11th? OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air
flight 115.
C₁₃: OK.
A₁₄: And you said returning on May 15th?
C₁₅: Uh, yeah, at the end of the day.
A₁₆: OK. There's #two non-stops ...#
C₁₇: #Act... actually #, what day of the week is the 15th?
A₁₈: It's a Friday.
C₁₉: Uh hmm. I would consider staying there an extra day til Sunday.
A₂₀: OK...OK. On Sunday I have ...

Properties of Human Conversation

- **Turns**
 - We call each contribution a “turn”
 - As if conversation was the kind of game where everyone takes turns.
- **Turn-taking issues**
 - When to take the floor?
 - When to yield the floor?
- **Interruptions**

Implications for Dialog Agents

- Barge-in
 - Allow the user to interrupt
- End-pointing
 - The task for a speech system of deciding whether the user has stopped talking.
 - Very hard, since people often pause in the middle of turns.

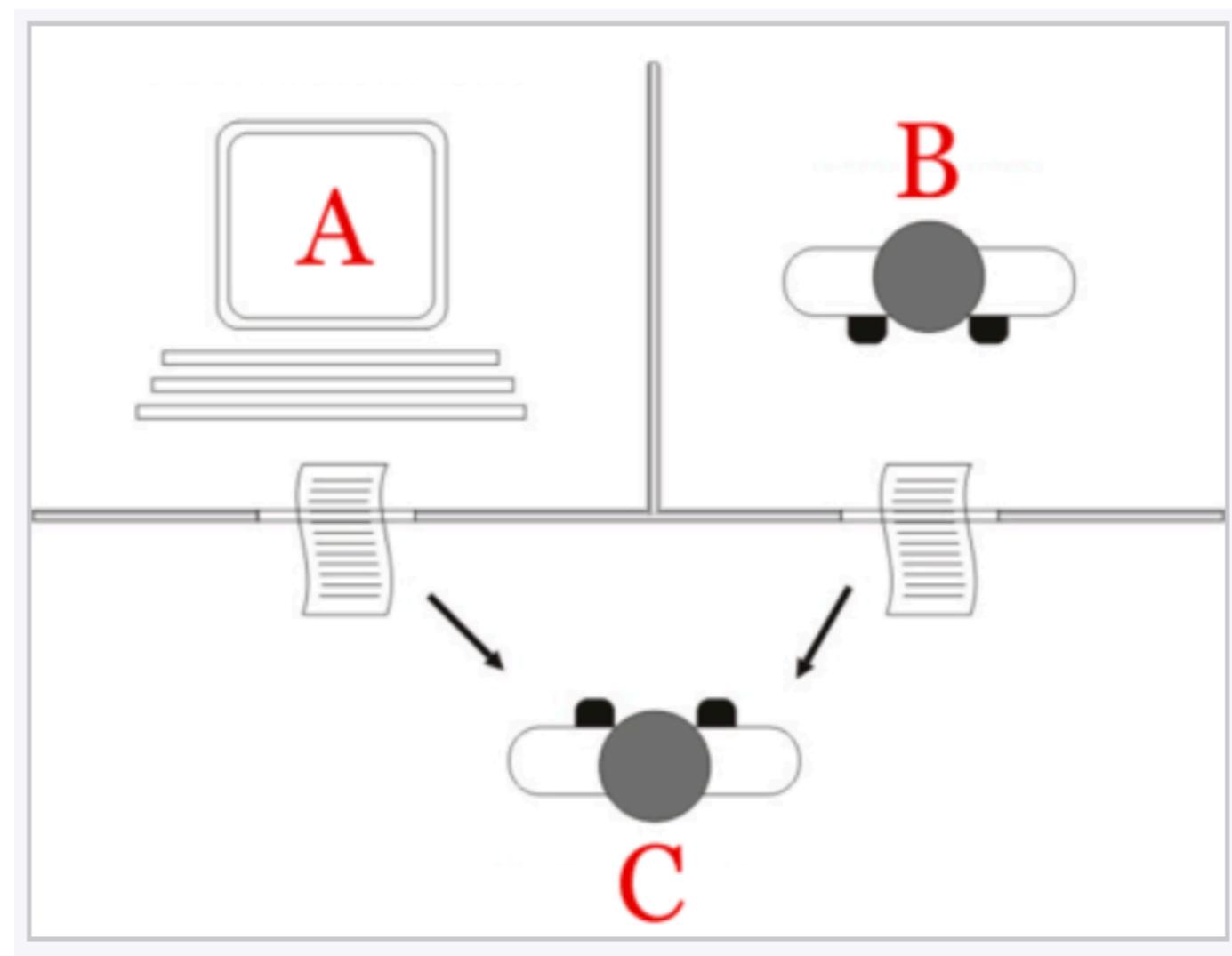
Language as Action



Each turn in a dialogue is a kind of action
Wittgenstein (1953) and Austin (1962)

Turing test

- Proposed by Alan Turing in 1950
- A test of a machine's ability to **exhibit intelligent behaviour** equivalent to, or indistinguishable from, that of a human.



Speech Acts (aka Dialogue Acts)

Bach and Harnish (1979)

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*)

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

...
You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU

My father is afraid of everybody.

WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER

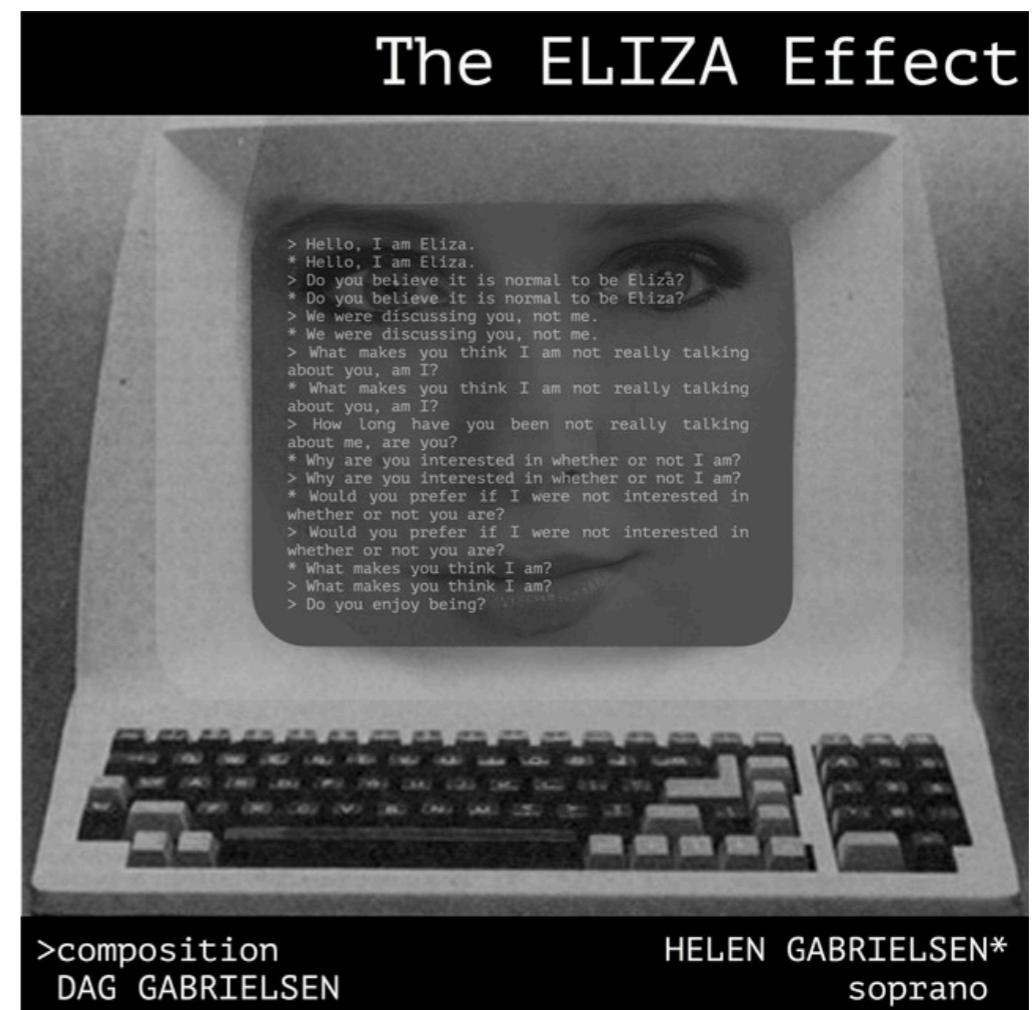
Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

Weizenbaum, Joseph. 1966, "ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine", Communications of the ACM 9 (1): 36-45

ELIZA Effect

- The **ELIZA effect**, in computer science, is the tendency to unconsciously assume computer behaviors are analogous to human behaviors; that is, [anthropomorphisation](#).



ELIZA's trick: be a Rogerian psychologist

- Draw the patient out by reflecting patient's statements back at them
- Rare type of conversation in which one can “assume the pose of knowing almost nothing of the real world”

Rogerian psychologist

(Person-Centered Therapy)

Patient: "I went for a long boat ride"

Psychologist: "Tell me about boats"

- You don't assume she didn't know what a boat is
- You assume she had some conversational goal
- Chatbots trying to pass the Turing test usually choose some such domain

Eliza Rules

- Rules are organized by **keywords**
- Each keyword has a pattern and a list of possible transform

Keyword: YOU

Pattern: (0 YOU 0 ME)

Transforms: (WHAT MAKES YOU THINK I 3 YOU)
(WHY DO YOU THINK I 3 YOU)

keyword pattern list of transforms

$$\begin{aligned} & (K ((D_1) (R_{1,1}) (R_{1,2}) \cdots (R_{1,m_1}))) \\ & ((D_2) (R_{2,1}) (R_{2,2}) \cdots (R_{2,m_2})) \\ & \vdots \\ & ((D_n) (R_{n,1}) (R_{n,2}) \cdots (R_{n,m_n}))) \end{aligned}$$

Keywords are ranked from specific to general

User: I know everybody laughed at me

“I” is very general:

I: (I *) -> (YOU SAY YOU 2)

ELIZA: YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU

“Everybody” is more specific and interesting

Everybody: (Everybody *) -> (WHO IN PARTICULAR ARE YOU THINKING OF)

ELIZA: WHO IN PARTICULAR ARE YOU THINKING OF?

Implementation: keywords stored with their rank

Everybody 5 (list of *transformation rules*)

I 0 (list of *transformation rules*)

Memory

```
(MEMORY MY  
  (0 MY 0 = LETS DISCUSS FURTHER WHY YOUR 3)  
  (0 MY 0 = EARLIER YOU SAID YOUR 3)  
  (0 MY 0 = DOES THAT HAVE ANYTHING TO DO WITH THE FACT  
THAT YOUR 3))
```

- Whenever “MY” is highest keyword
 - Randomly select a transform on the MEMORY list
 - Apply to sentence
 - Store on a (first-in-first-out) queue
- Later, if no keyword matches a sentence
 - Return the top of the MEMORY queue instead

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 \leftarrow Apply the transform in *r* to *sentence*

if *w* = ‘my’

future \leftarrow Apply a transformation from the ‘memory’ rule list to *sentence*

Push *future* onto memory stack

else (no keyword applies)

either

response \leftarrow Apply the transform for the NONE keyword to *sentence*

or

response \leftarrow Pop the top response from the memory stack

return(*response*)

PARRY passes the Turing test in 1972

The first system to pass a version of the Turing test

Psychiatrists couldn't distinguish interviews with PARRY from (text transcripts of) interviews with people diagnosed with paranoid schizophrenia

- Colby, K. M., Hilf, F. D., Weber, S., and Kraemer, H. C. (1972). Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. *Artificial Intelligence* 3, 199–221.

Microsoft Xiaoice

(Zhou et al. 2020)

- Microsoft's chatbot responds mainly by extracting turns that humans have said in past conversations.

[人类] 小冰你好

[小冰] 你好呀，你头像好有趣，感觉在哪见过

[人类] 是吗

[小冰] 提不是哪部动画里的呀

[人类] 你怎么知道的

[小冰] 你猜

[人类] 难道你是真人吗？

[小冰] 这个，你和我多聊聊就知道了

[人类] 你不是人工智能机器人么

[小冰] 是又怎样，我不会冷冰冰的对待你的

[人类] 你都有什么功能啊

[小冰] 我有200多个skills，来，说出你的需求

[Human] Hello, XiaoIce.

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

[Human] Really?

[XiaoIce] Is it a comic character?

[Human] How do you know that?

[XiaoIce] Guess.

[Human] Are you human?

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

[Human] Aren't you an AI chatbot?

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

[Human] What functions do you have?

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

Meta BlenderBot

(Roller et al. 2020)

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.

Google LaMDA

(Thoppilan et al 2022)

- LLM-based Chatbot
- “Google AI Researcher was placed on administrative leave after going public with claims that LaMDA was sentient.”

OpenAI ChatGPT

(Long et al. 2022)

- Reinforcement learning with human feedback

JU

How do I make an HTTP request in Javascript?



You can make an HTTP request in JavaScript using the built-in `fetch` API or the older `XMLHttpRequest` (XHR) API.

Using the Fetch API:

The Fetch API is a newer and simpler way of making HTTP requests in JavaScript. Here's an example of how to use the Fetch API to make an HTTP GET request:

javascript

Copy code

```
fetch('https://example.com/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

JU

Can you recommend some good restaurants in Madison downtown?



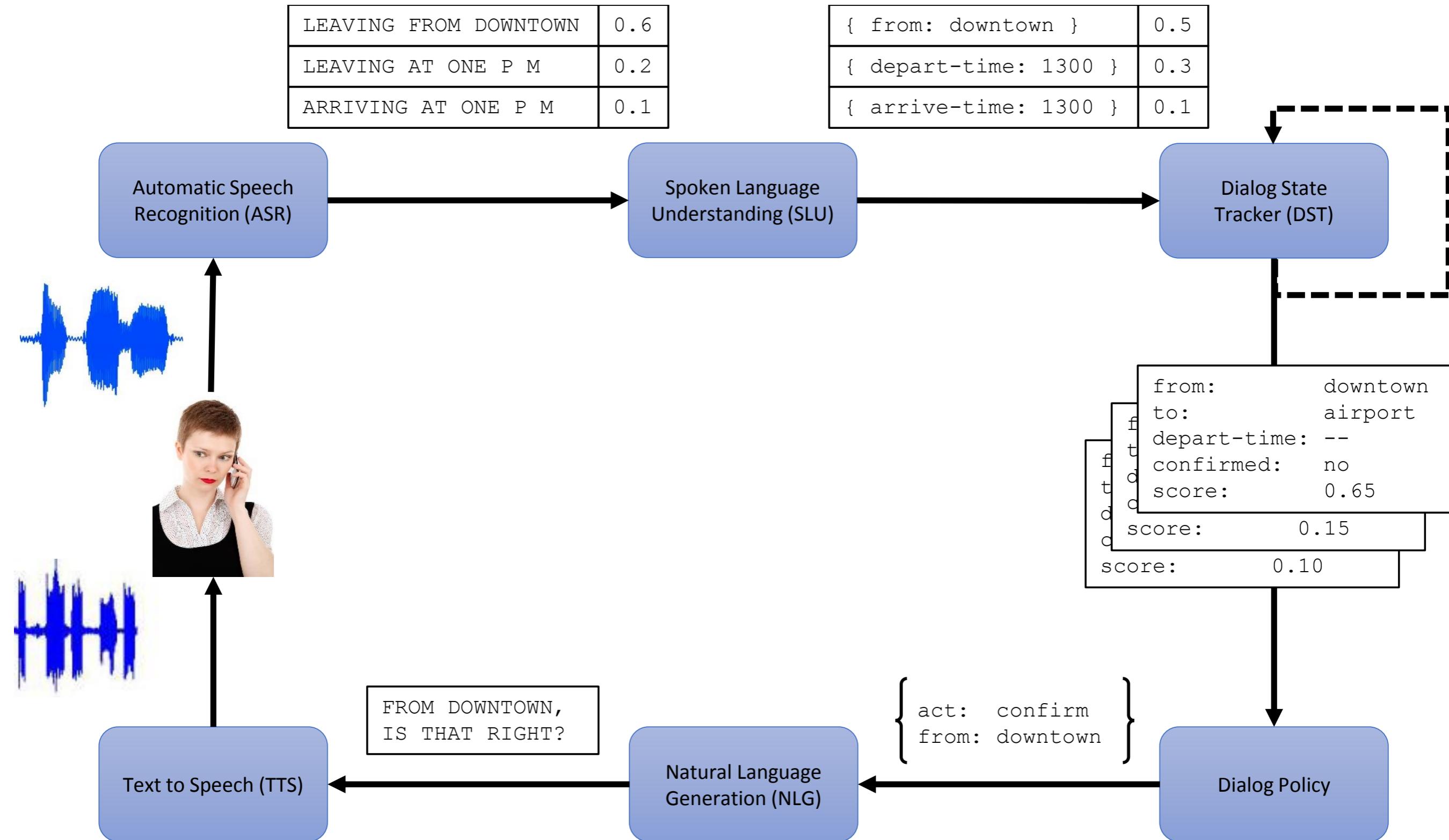
Certainly! Madison has a great selection of restaurants in its downtown area. Here are a few options that are highly recommended:

1. Graze - This restaurant is known for its farm-to-table approach and has a great selection of brunch and dinner options.
2. L'Etoile - This upscale restaurant offers a sophisticated dining experience with a focus on seasonal and locally-sourced ingredients.
3. Heritage Tavern - This restaurant features contemporary American cuisine with an emphasis on artisanal meats and locally-sourced produce.

Task-oriented Dialog

Pipeline Dialog System

(Williams et al. 2016)



Components in a ToD architecture

- **NLU:** extracts slot fillers from the user's utterance using machine learning
- **Dialogue state tracker (DST):** maintains the current state of the dialogue (user's most recent dialogue act, set of slot-filler constraints from user), sometimes merged in the **NLU**.
- **Dialogue policy (DP or DM):** decides what the system should do or say next
- **NLG:** produce more natural, less templated utterances

NLU

Natural language understanding in dialog involves several key tasks:

- Intent prediction: what is the user's intent/goal
- Slot filling: what are the slot values (e.g., what is the time)
- **Dialog State tracking (DST):** track user information/goals throughout the dialog

NLU

Natural language understanding in dialog involves several key tasks:

- DialoGLUE [Mehri et al. 2020]
- Intent prediction: ATIS, SNIPS, Banking77, CLINC150, HWU64
- Slot filling: ATIS, SNIPS, DSTC8-SGD, Restaurant8k
- State tracking: MultiWOZ (2.X)

Slot filling

- Machine learning classifiers to map words to semantic frame-filters
- 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

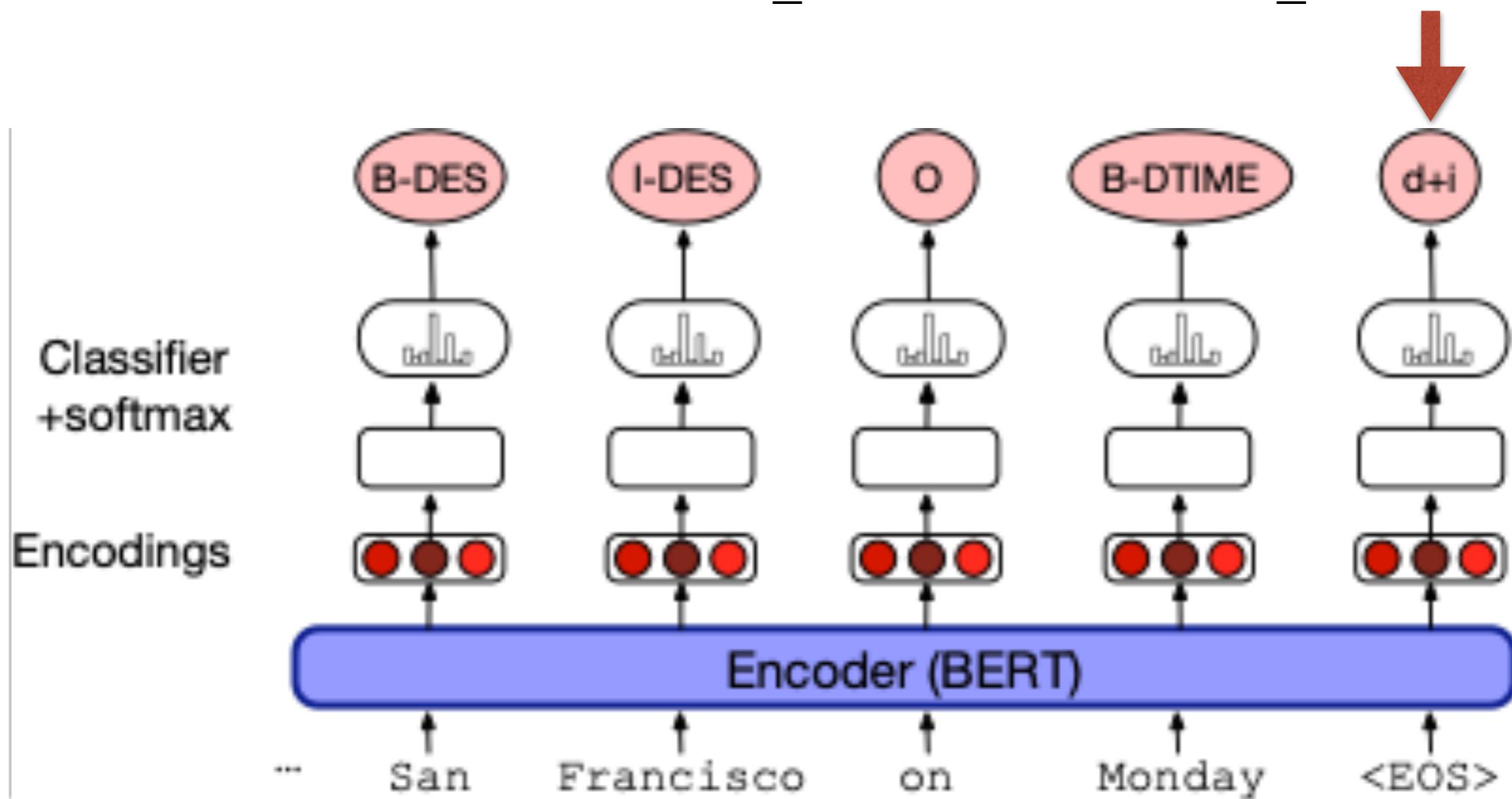
- The **BIO tagging** paradigm
- Idea: Train a classifier to label each input word with a tag that tells us what slot (if any) it fills

0 0 0 0 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"



GenSF

(Mehri and Eskenazi. 2021)

- Convert slot filling task as a response generative task.
- Asking the slot information using a template.
- Make the fine-tuning task close to the pre-training task (LM)

Requested Slots

first name

last name

Utterance

Laurice Hoisl

Slots

date

time

first name

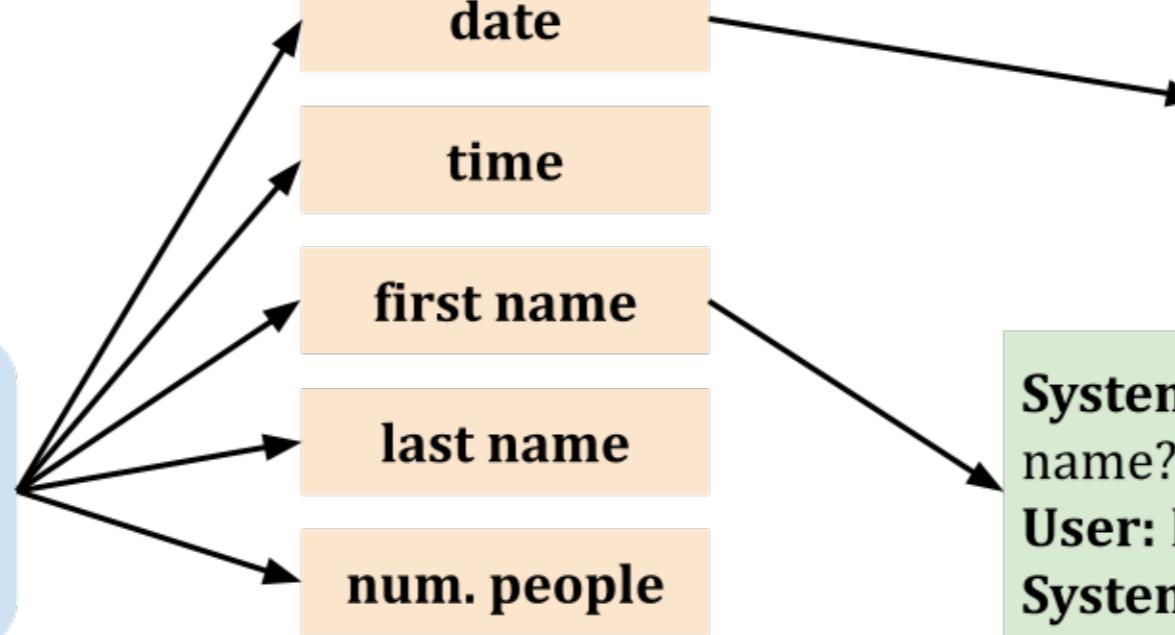
last name

num. people

Dialog

System: What is the first name, last name?
User: Laurice Hoisl
System: Ok, the date is __

System: What is the first name, last name?
User: Laurice Hoisl
System: Ok, the first name is __



Results on Restaurant8k

Fraction	Span-BERT	ConVEx	GenSF
1 (8198)	93.1	96.0	96.1
1/2 (4099)	91.4	94.1	94.3
1/4 (2049)	88.0	92.6	93.2
1/16 (512)	76.6	86.4	89.7
1/128 (64)	30.6	71.7	72.2

Dialog State Tracking

- Maintain the current state of the dialog—update the state from the current user turn incrementally

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

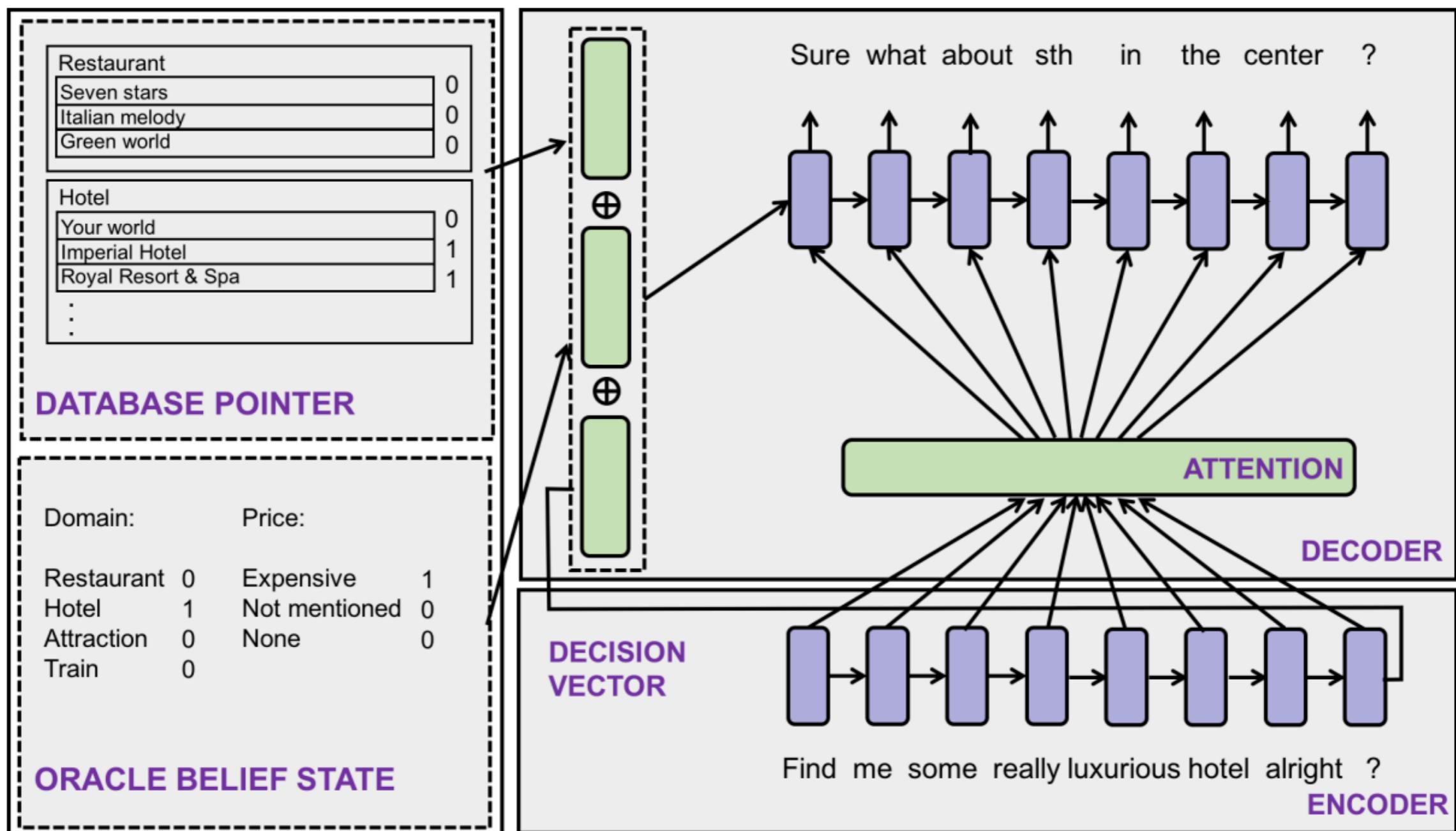
Task-Oriented Response Generation

ToD response generation module should reflect below factors:

- Must understand the **dialog context**
- Must track **belief state** over dialog context
- Often need to interpret structured **database output**
- Must follow task-specific **dialog policy**
- Must generate **fluent, coherent** natural language responses

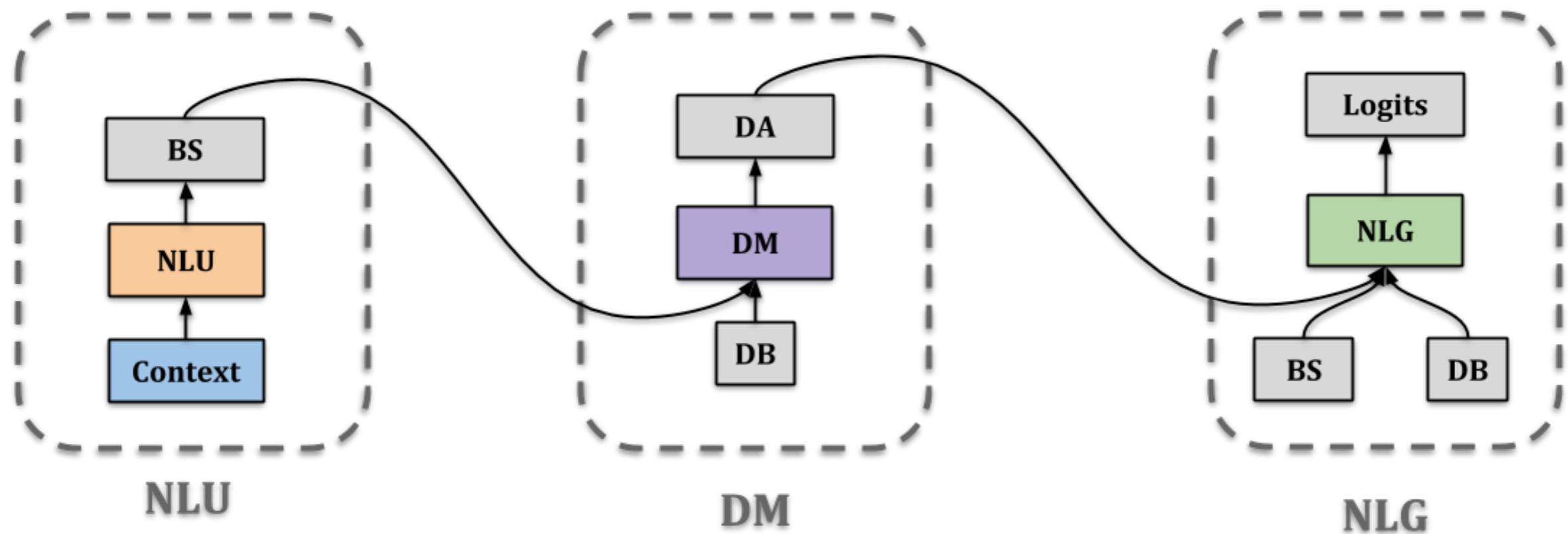
Seq2Seq with Attention

(Budzianowski et al. 2018)



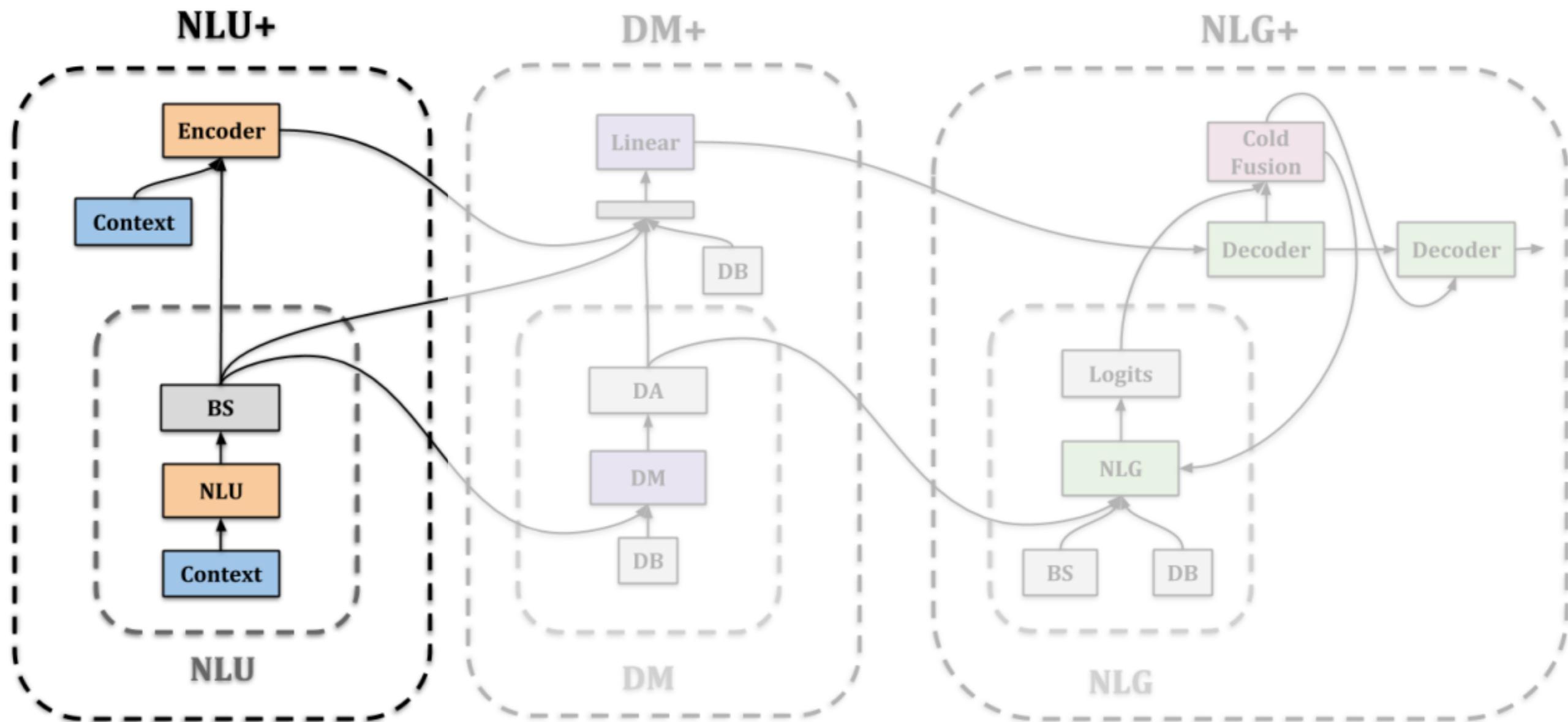
Dialog Modules

Start with **pre-trained** neural dialog modules



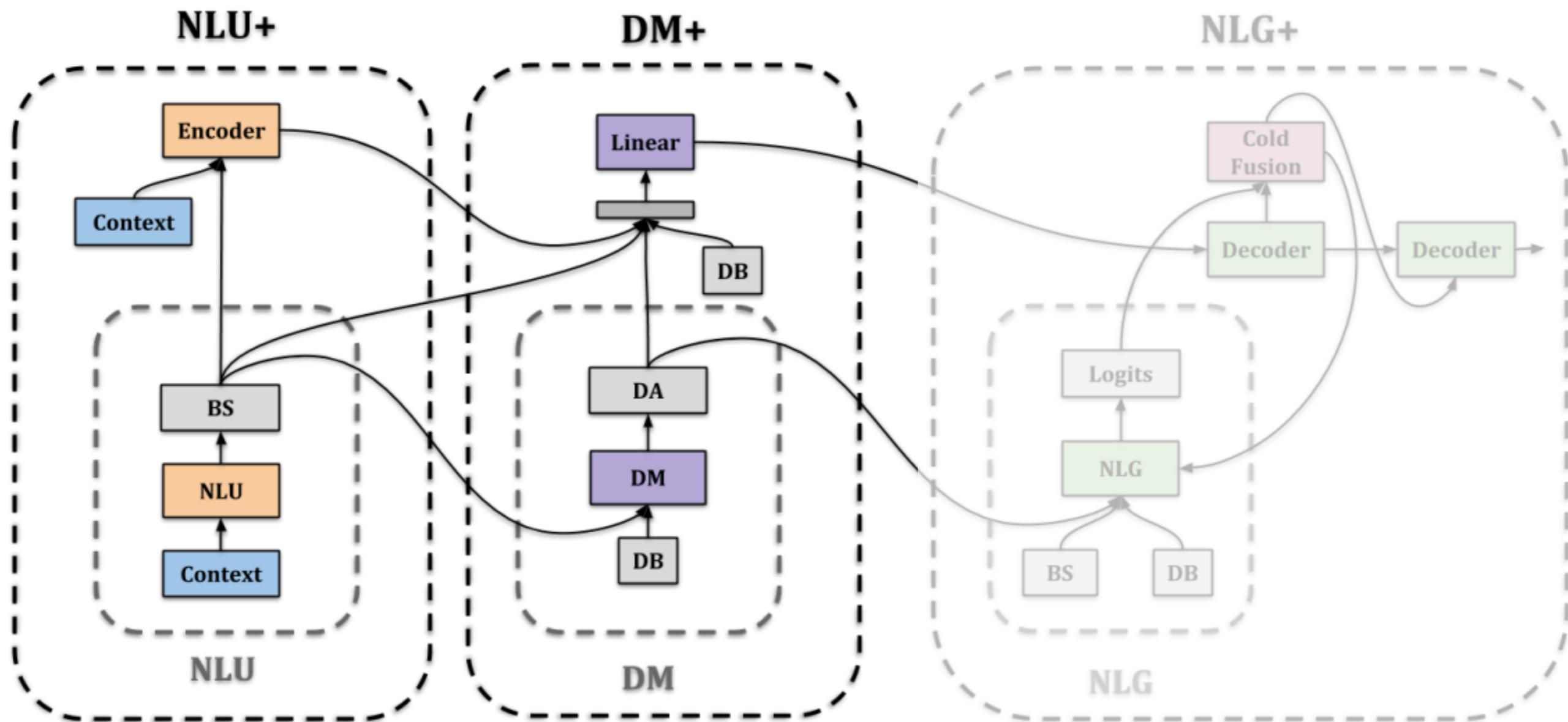
Structured Fusion Networks

(Mehri et al. 2019)



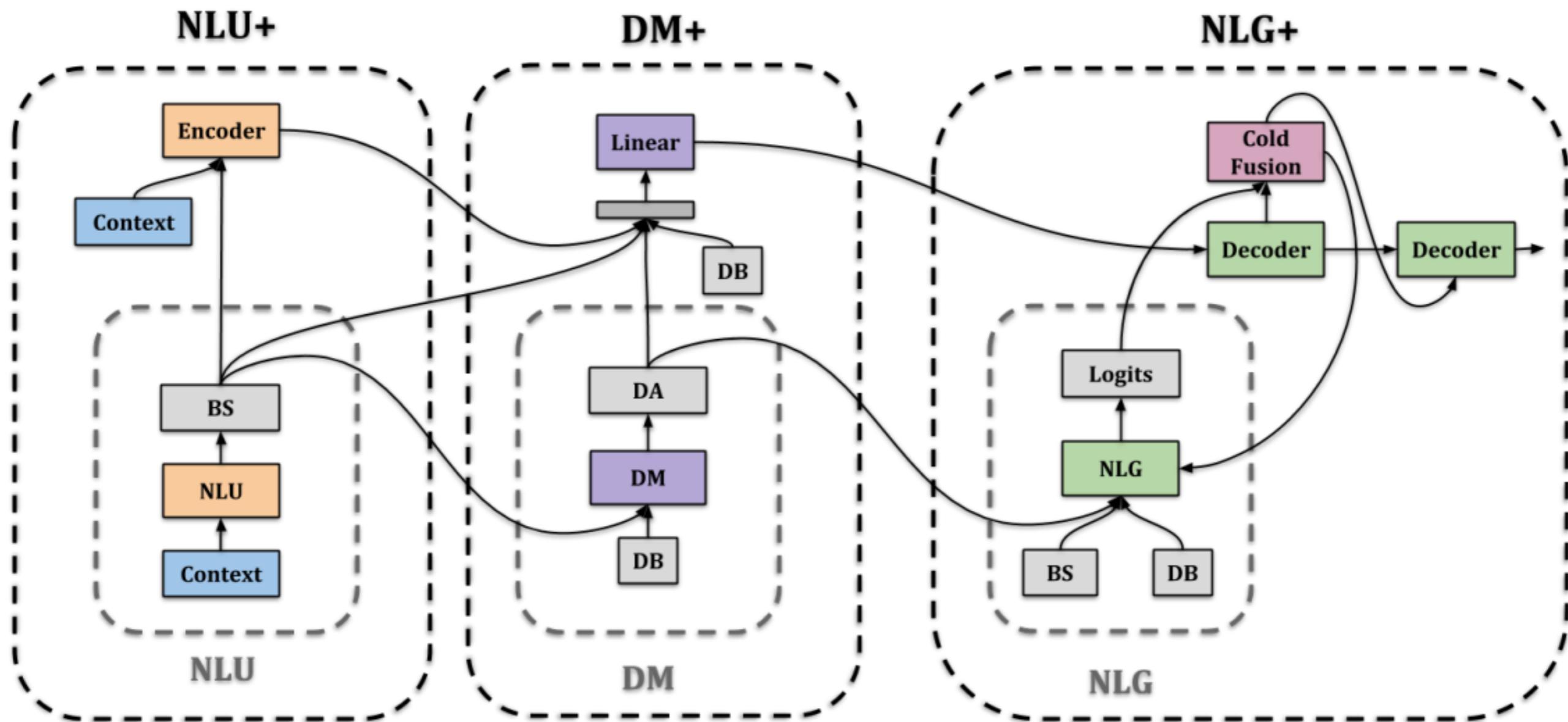
Structured Fusion Networks

(Mehri et al. 2019)



Structured Fusion Networks

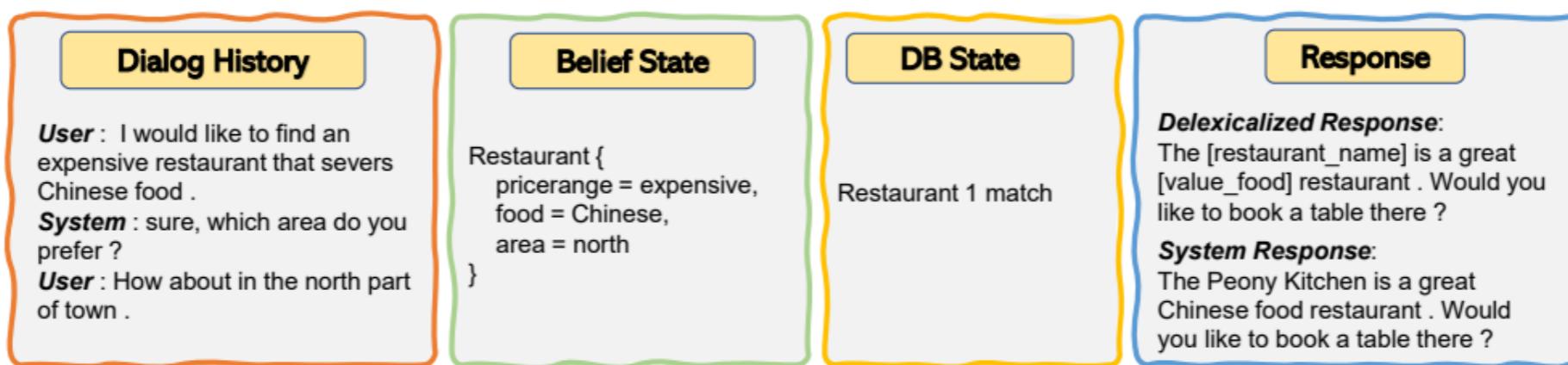
(Mehri et al. 2019)



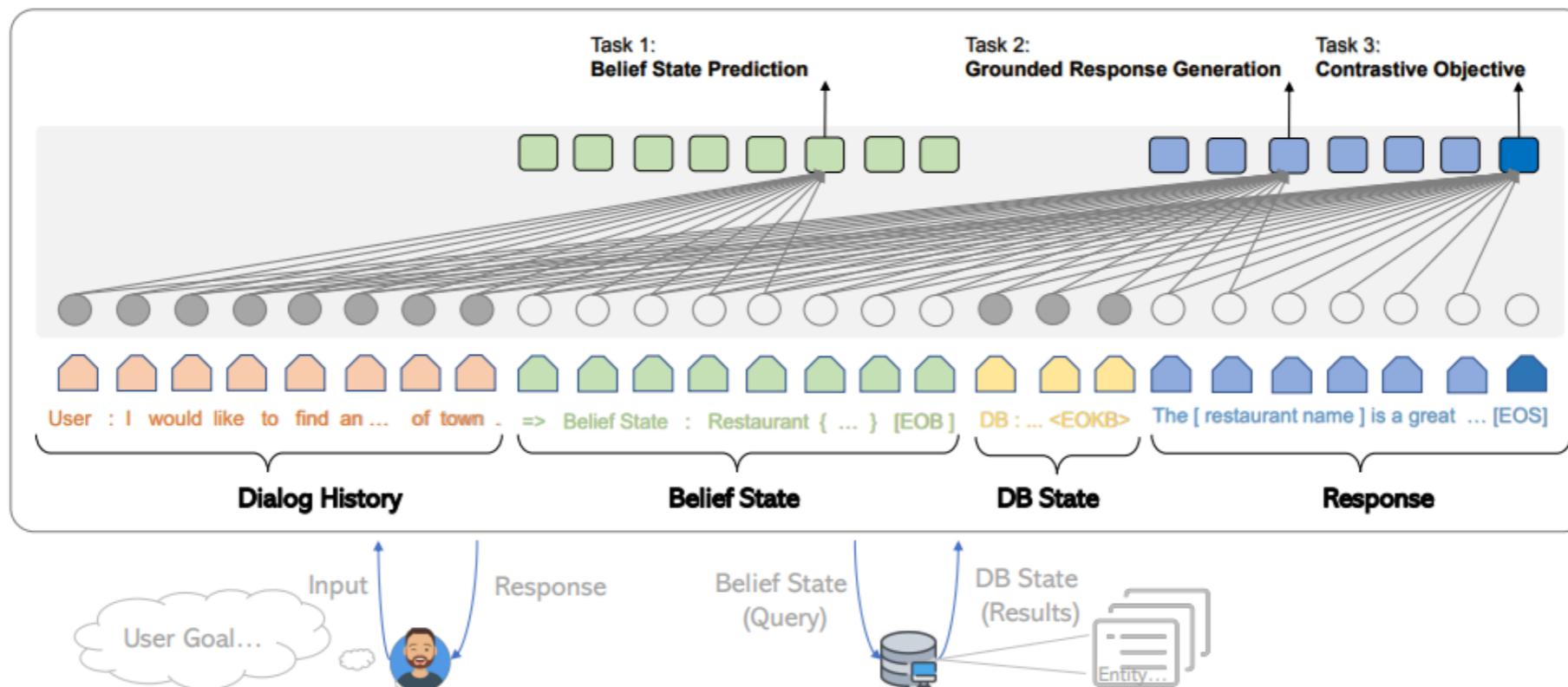
Soloist

(Peng et al. 2020)

- Pre-train a LM on heterogeneous dialog corpora
- Adapt to new tasks with a handful of task-specific dialogs via machine teaching



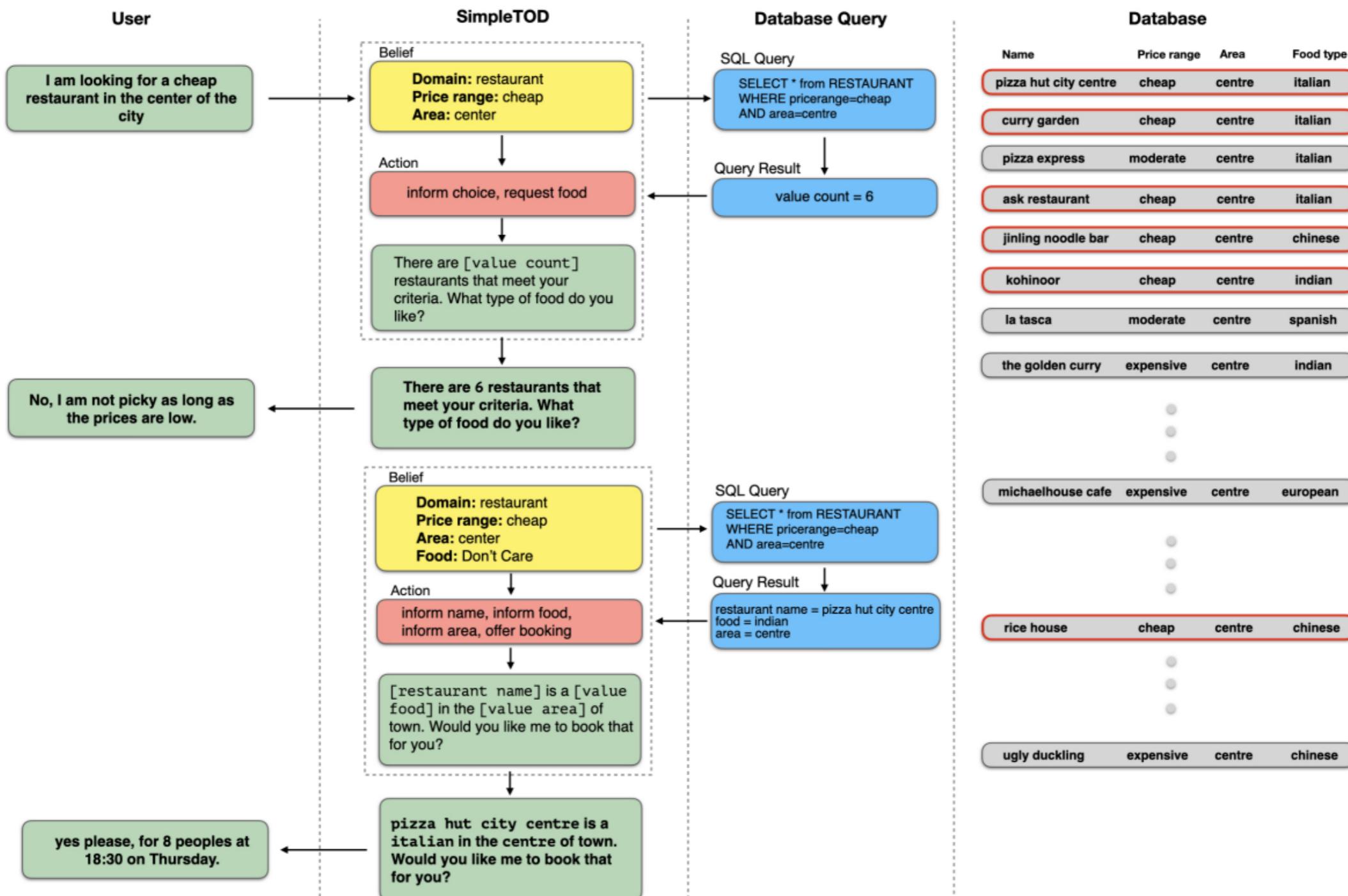
(b) Example snippets for the items compounding the input of Soloist model.



SimpleTOD

(Hosseini-Asl et al. 2020)

- Use a single LM (GPT2) to generate all outputs given the dialog context and retrieved database search results.



Open-domain Dialog (chit-chat)

Open-Domain Response Generation

Open-domain dialog systems must engage in chit-chat with a user

- DailyDialog [Li et al. 2017]
- PersonaChat [Zhang et al. 2018]
- Topical-Chat [Gopalakrishnan et al. 2019]
- Wizards of Wikipedia [Dinan et al. 2018]
- Empathetic Dialogs [Rashkin et al. 2019]

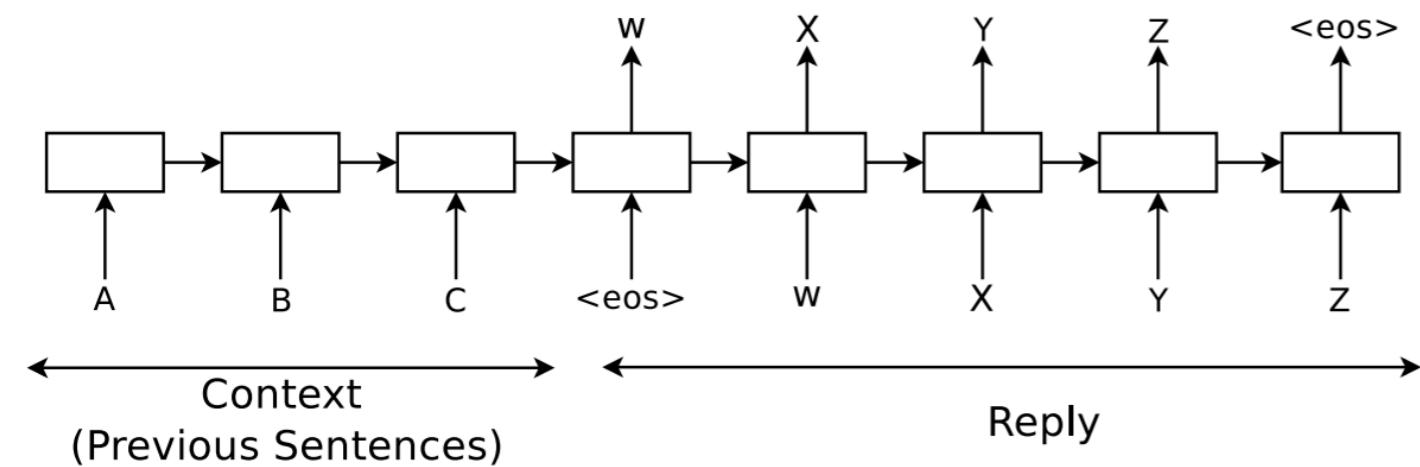
Open-Domain Response Generation

Open-domain dialog systems must engage in chit-chat with a user

- Must understand the **dialog context**
- Must be able to discuss a variety of **topics**
- Must generate natural language **responses**
- Must generate **engaging/interesting** responses
- Must demonstrate **common sense reasoning**

Seq2Seq

(Vinyals and Le. 2015)



Human: *what is the purpose of life ?*

Machine: *to serve the greater good .*

Human: *what is the purpose of living ?*

Machine: *to live forever .*

Human: *what is the purpose of existence ?*

Machine: *to find out what happens when we get to the planet earth .*

Human: *where are you now ?*

Machine: *i 'm in the middle of nowhere .*

Human: *what is the purpose of dying ?*

Machine: *to have a life .*

Human: *what is the purpose of being intelligent ?*

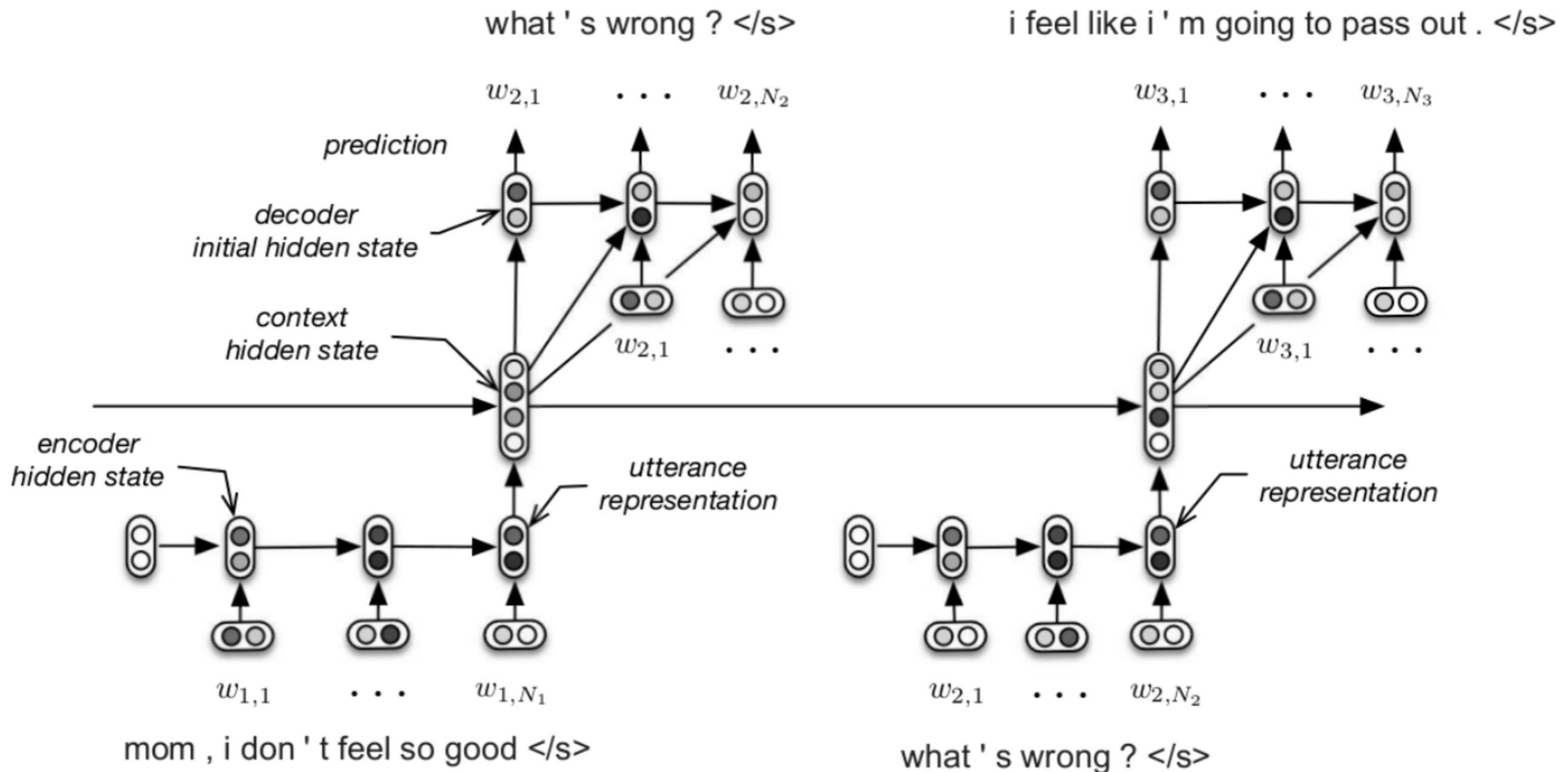
Machine: *to find out what it is .*

Human: *what is the purpose of emotions ?*

Machine: *i don 't know .*

HR-ED

(Serban et al. 2016)



Persona-Conditioned Models

(Zhang et al. 2018)

- To make open-domain chit-chat dialog models more consistent and engaging, condition them on a persona

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

[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!

[PERSON 2:] Good choice. Do you watch Game of Thrones?

[PERSON 1:] No, I do not have much time for TV.

[PERSON 2:] I usually spend my time painting: but, I love the show.

Persona-Conditioned Models

(Zhang et al. 2018)

Persona 1	Persona 2
I am engaged to be married	I am a writer
I used to play sports in high school	I write romance novels
My favorite sport is soccer	I live on a faraway island
I am a college graduate	I am an introvert
I love watching sports on TV	I have two cats

[PERSON 1:] I just got done playing soccer. What have you been up to?

[PERSON 2:] I do not have pets. I spend my free time writing. what about you?

[PERSON 1:] Just graduated from college. Looking for work.

[PERSON 2:] yes I graduated, in writing and poetry

[PERSON 1:] Have you had anything published?

[PERSON 2:] I mainly write fiction novels.

[PERSON 1:] Do you like to watch sports?

[PERSON 2:] do you like kings of leon my favorite by them is use somebody

[PERSON 1:] Are you married? I will be married soon.

[PERSON 2:] haha, no time. I have got a novel to finish.

[PERSON 1:] What is your novel about?

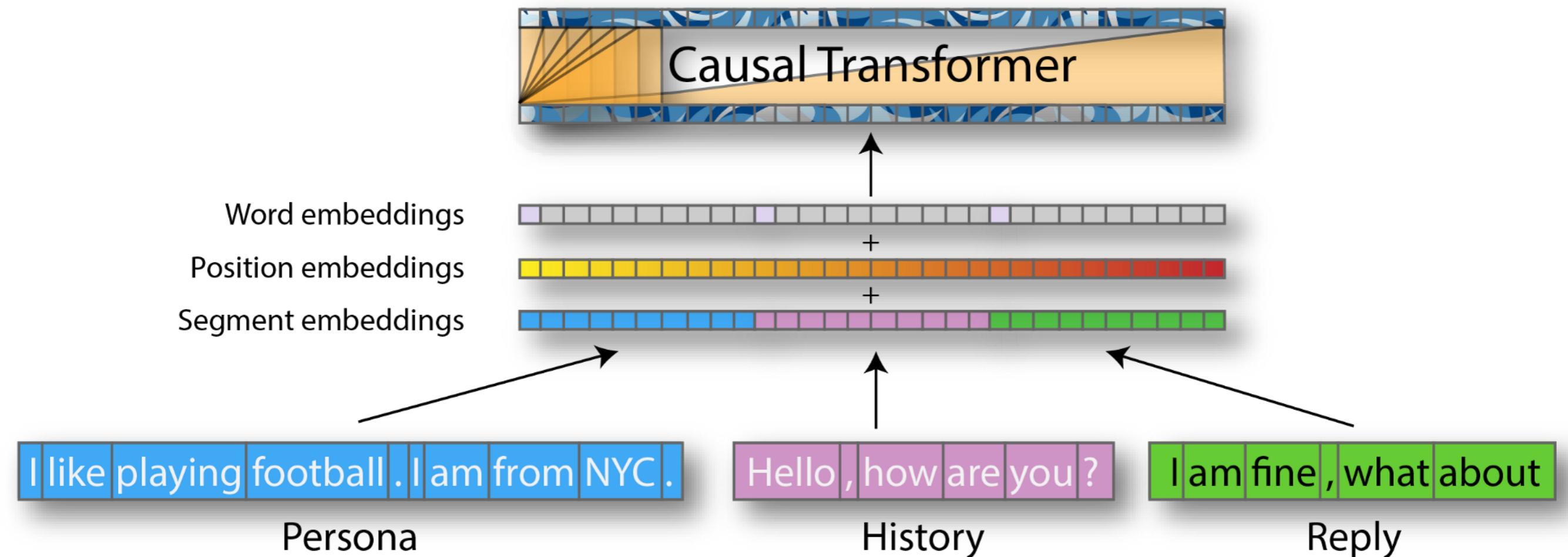
[PERSON 2:] it is a love story. I really enjoy writing. do you write or read?

[PERSON 1:] Like to read. Spend most time watching football.

[PERSON 2:] yes I do I am and author for a living

Transfer-Transfo

(Wolf et al. 2019)



DialoGPT

(Zhang et al. 2019)

- Continue pre-training GPT-2 on **conversations from Reddit**
 - Filter long utterances
 - Filter non-English utterances
 - Filter URLs
 - Filter toxic comments
- Train on 147M dialog instances (1.8B words)
- “Human-level” response generation ability

Open-Domain Response Generation

- **Knowledge-grounded** response generation
- **Persona-grounded** response generation
- **Negotiation/persuasive** dialog
- **Commonsense** dialog
- Conversational **QA**

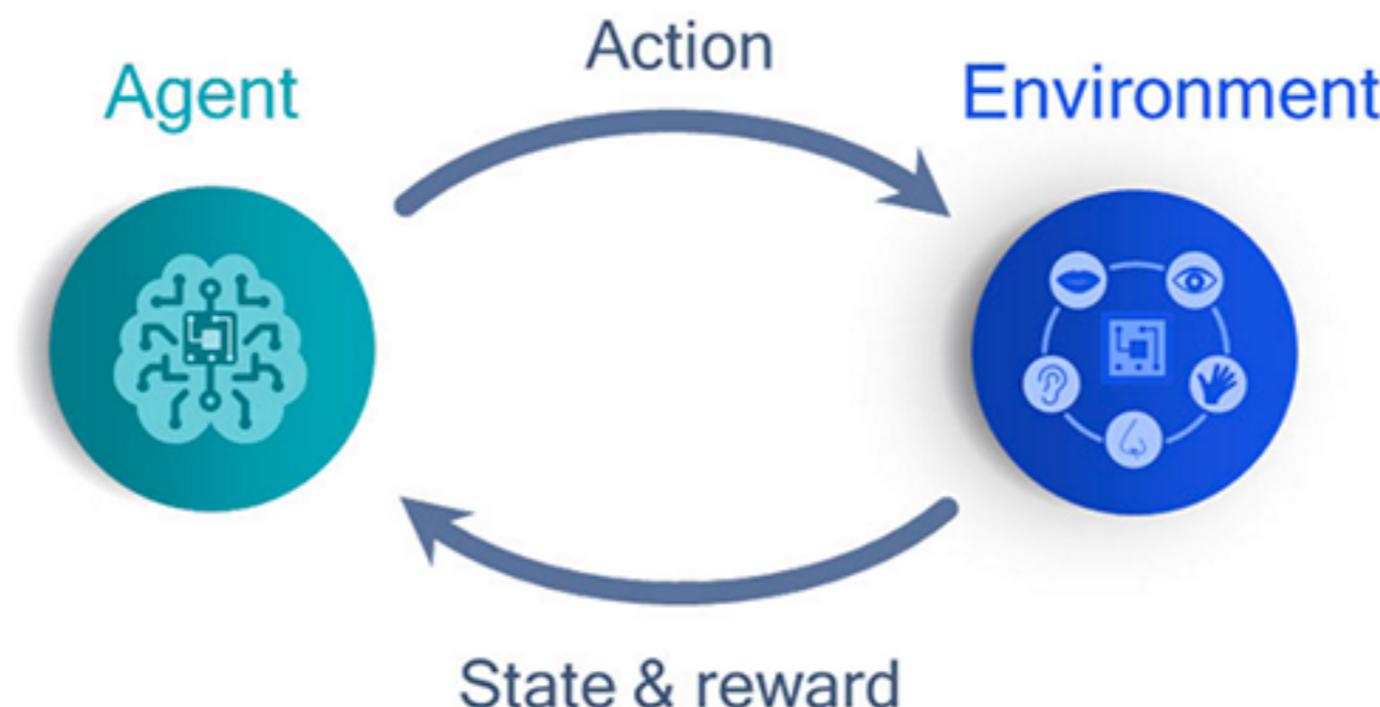
LLM-based Chatbot (ChatGPT)

LLM as Implicit Multi-task Chatbot

- Both **ToD** and **Chit-chat converge** to use **LLMs** as a backbone over the recent years.
- Compared with **pipeline ToD**, LLMs do not explicitly decouple multiple tasks, but instead convert all ToD tasks into a sequence generation problem.
 - **ChatGPT Plugin**: access to external database
- For **Chit-chat**, we have seen several studies using LLMs with prompts (e.g., persona description) even prior to ChatGPT.

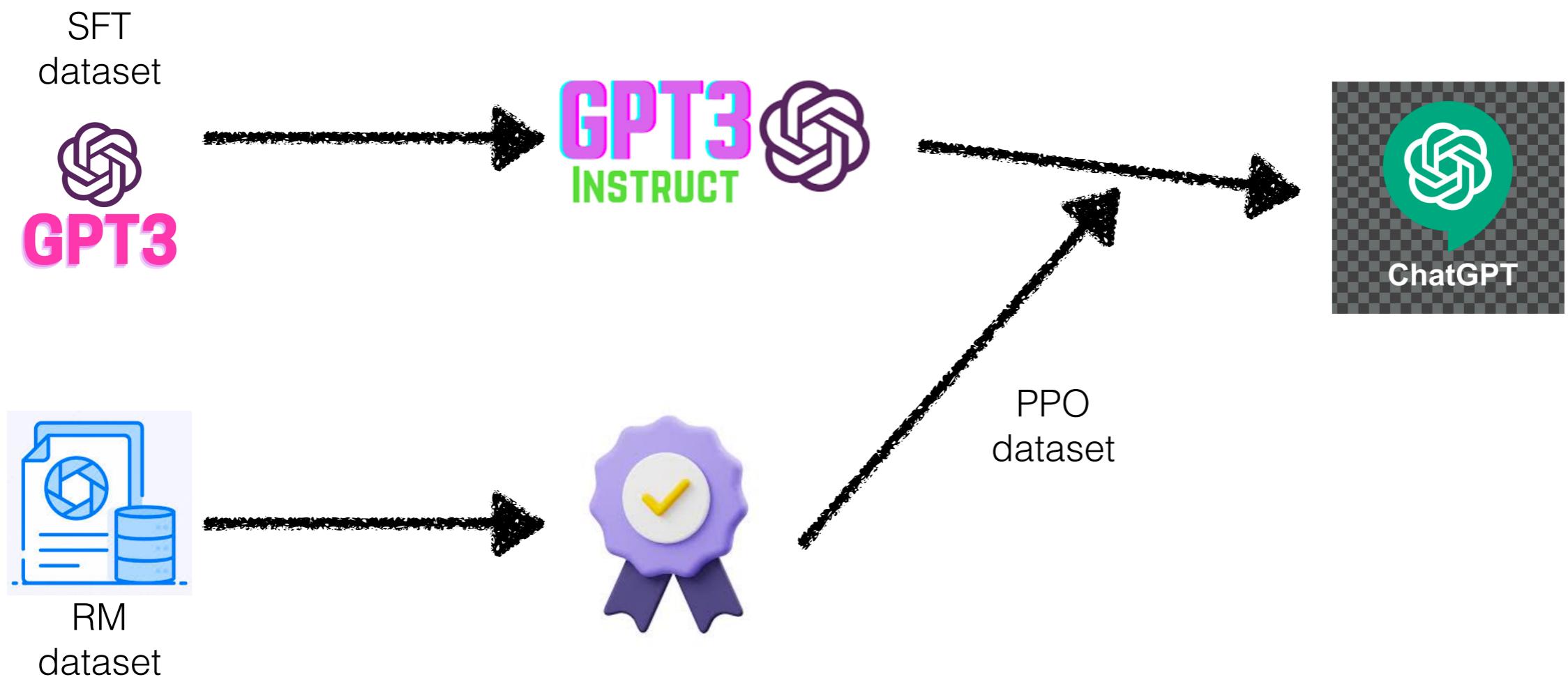
LLMs to Chatbot

- LLMs without any environment reward are still hard to get meaningful responses during human-computer interaction
- **Challenge:** simulating an environment in an online setting is costly.
- **Solution:** train a reward model to estimate environment rewards.



ChatGPT

- Supervised fine-tuning (SFT)
- Training reward modeling (RM)
- Reinforcement learning with human feedback (RLHF)



Dataset Collection

- **Prompt dataset collections:** each prompt defines a query task
 - **Plain:** ask labelers to come up with an arbitrary task w/ sufficient diversity
 - **Few-shot:** ask labelers to come up with an instruction, and multiple Q/R pairs for the instruction
 - **User-based:** OpenAI collected a number of use-cases stated in waitlist applications to the OpenAI API. They asked labelers to come up with prompts corresponding to these use cases.

Dataset Collection

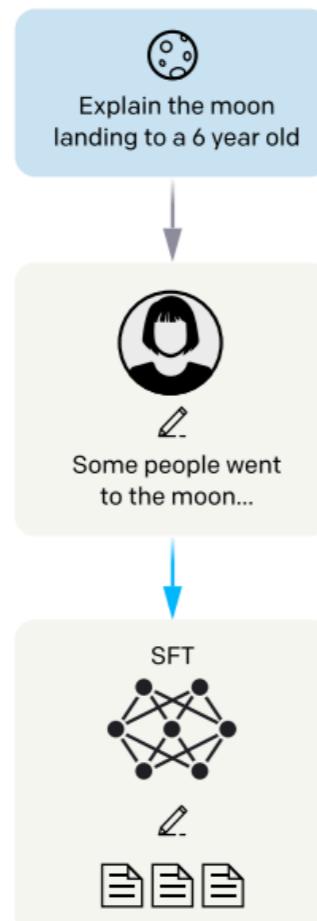
- **Three dataset collections:**
 - **SFT dataset:** ask labelers to write responses to a given prompt
 - **RM dataset:** ask labelers to provide rankings of model outputs used to train the reward model.
 - **PPO dataset:** the dataset collected during RLHF without any human labels — using RM to provide rewards.

Learning from Human Feedback

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

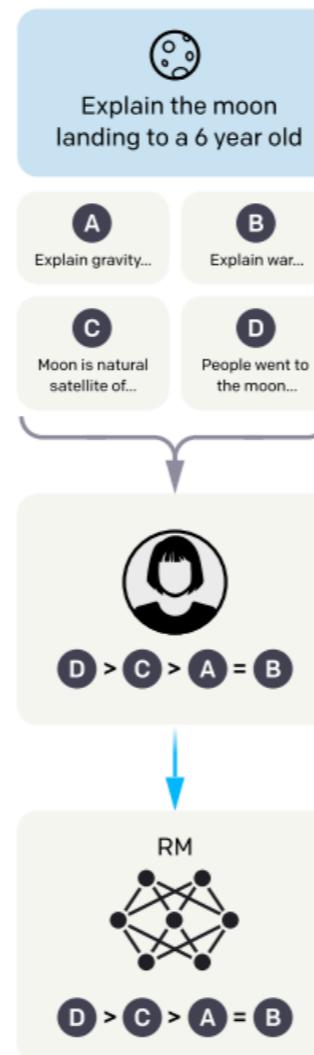


A labeler demonstrates the desired output behavior.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



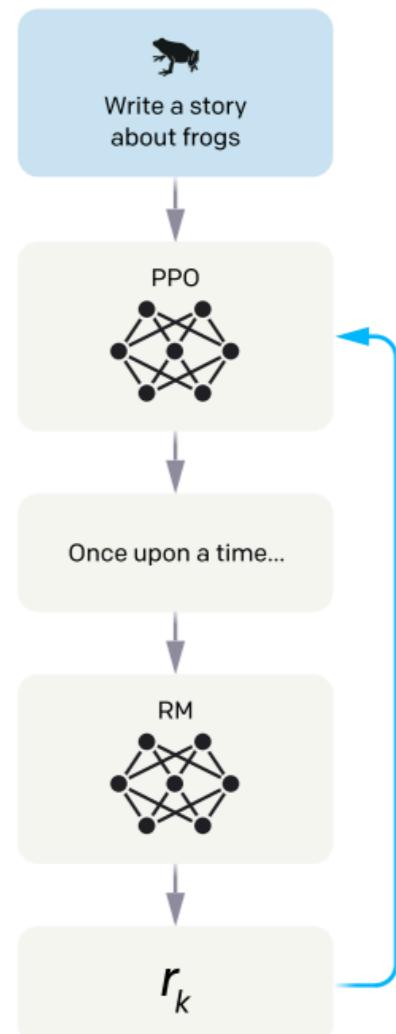
A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



The policy generates an output.

The reward model calculates a reward for the output.

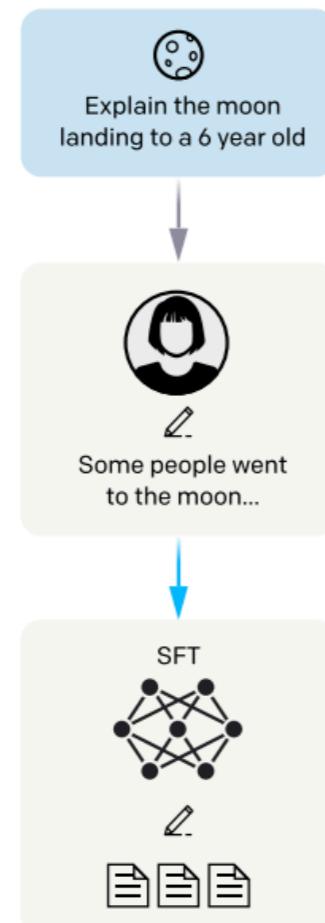
The reward is used to update the policy using PPO.

Learning from Human Feedback

Step 1

Collect demonstration data,
and train a supervised policy.

A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.

This data is used
to fine-tune GPT-3
with supervised
learning.

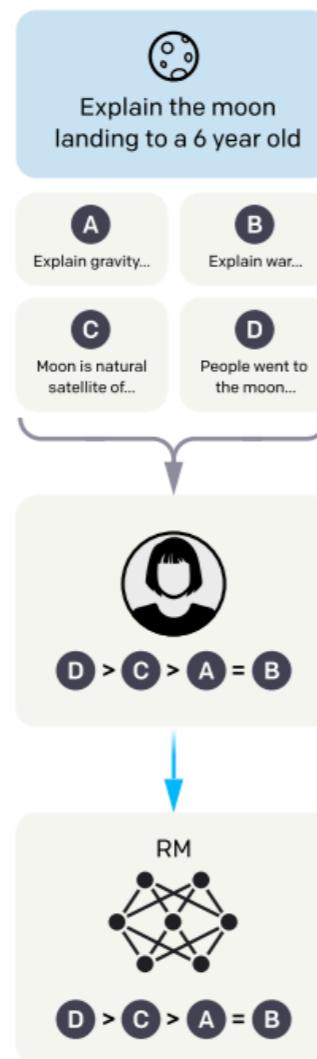
- **Supervised Fine-tuning (SFT):** Fine-tune a pretrained GPT-3 model on labeled data
- Use traditional cross-entropy with teaching forcing to fine-tune GPT-3

Learning from Human Feedback

Step 2

**Collect comparison data,
and train a reward model.**

A prompt and
several model
outputs are
sampled.



- **Reward Modeling (RM):** Train a RM to rank K responses ($K=4\sim 9$) to a prompt
- This produces $\binom{K}{2}$ combinations of pairwise comparison
- Pairwise ranking loss:

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} [\log (\sigma (r_\theta(x, y_w) - r_\theta(x, y_l)))]$$

This data is used
to train our
reward model.

Learning from Human Feedback

Step 3

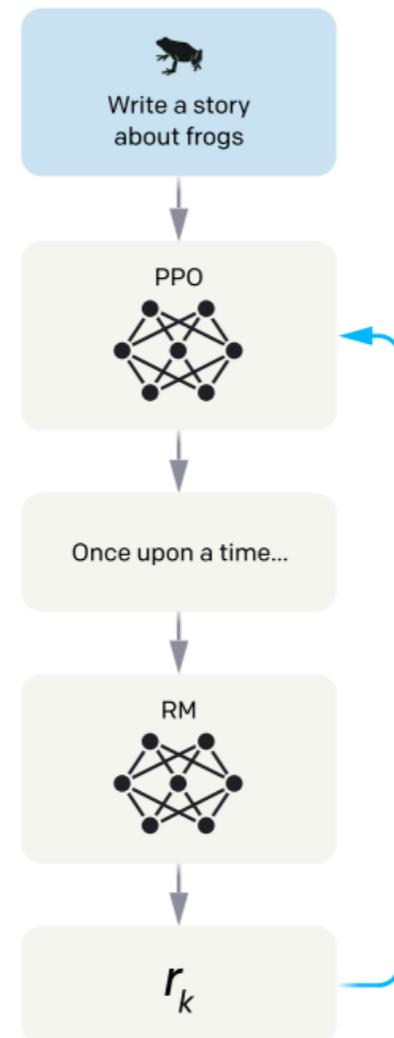
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



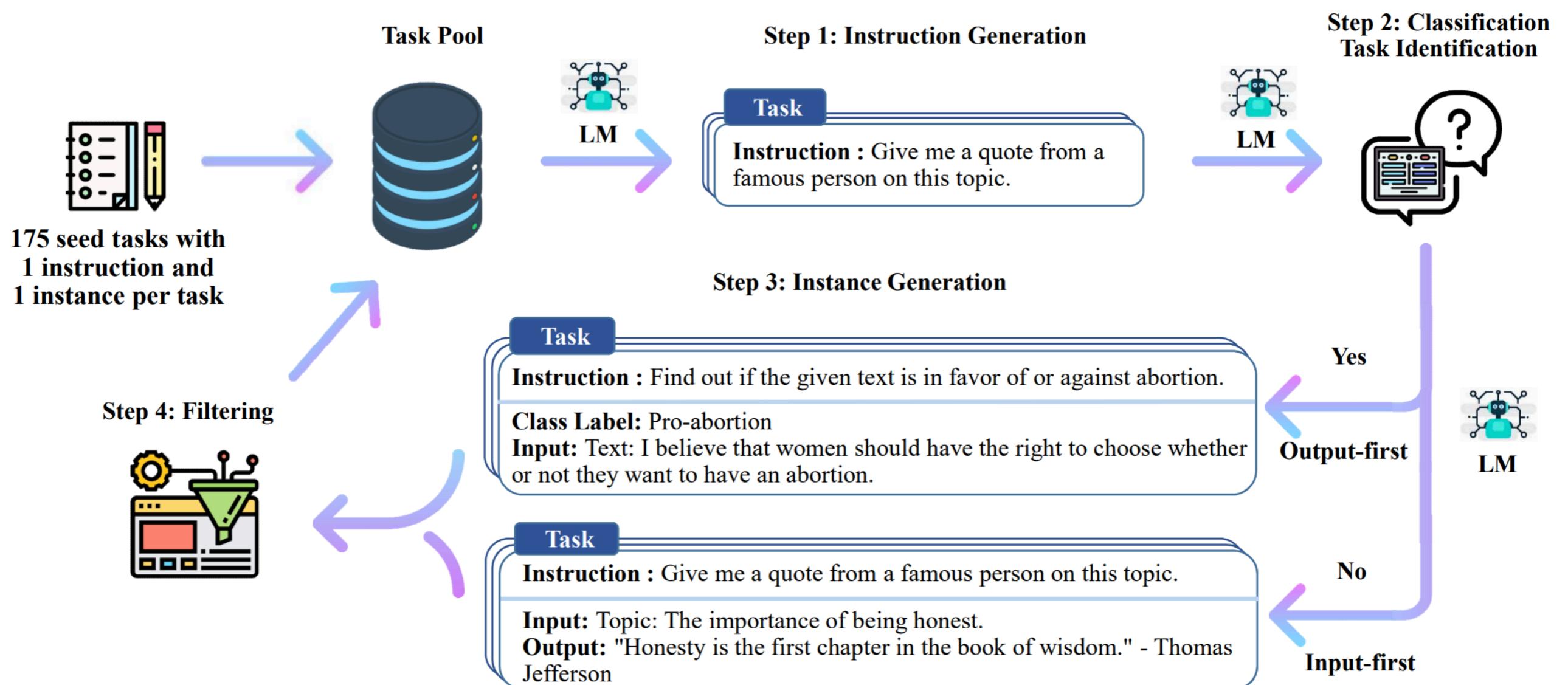
- **Reinforcement Learning (RL):** Train a RL policy model initialized from the SFT policy model
- Using PPO algorithm (Schulman et al. 2017)

$$\text{objective}(\phi) = \frac{E_{(x,y) \sim D_{\pi_\phi^{\text{RL}}}} [r_\theta(x, y) - \beta \log (\pi_\phi^{\text{RL}}(y | x) / \pi^{\text{SFT}}(y | x))] +}{\gamma E_{x \sim D_{\text{pretrain}}} [\log(\pi_\phi^{\text{RL}}(x))]}$$

LM pertaining on public NLP datasets RL (add a per-token KL penalty from SFT model)

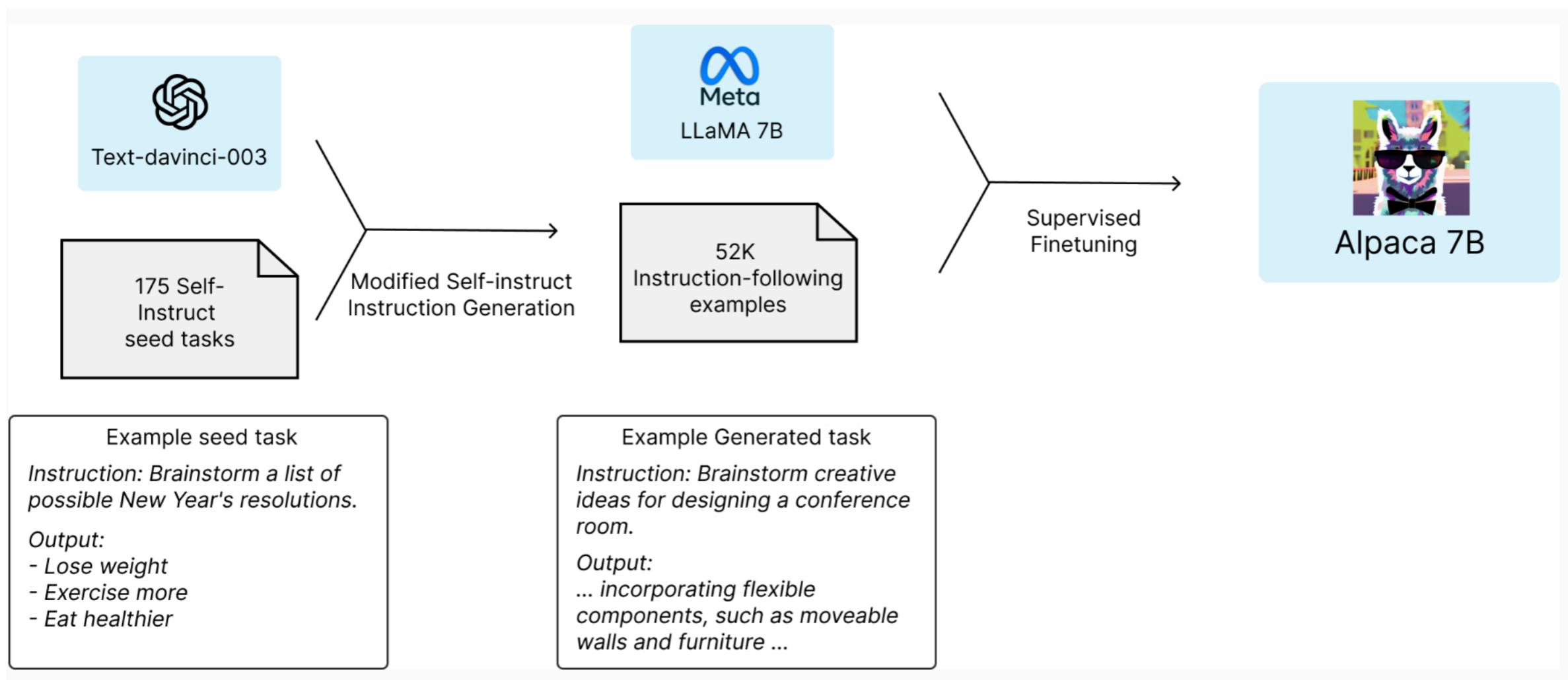
Self-instruction

- Distill information from LLM
- Ask LLMs to provide instructions



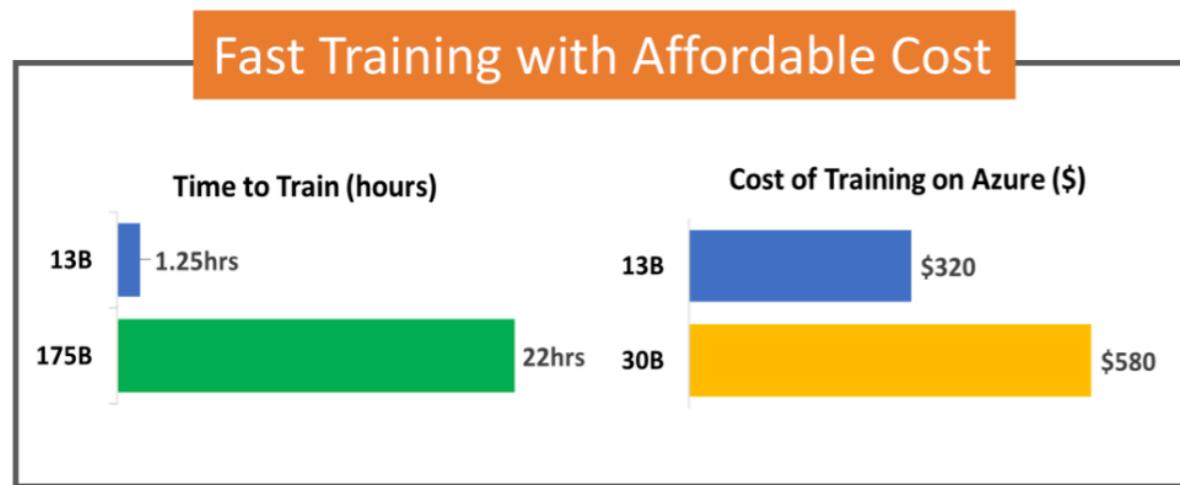
Stanford Alpaca

- Use self-instruct to get prompt-response pairs
- Use supervised fine-tuning

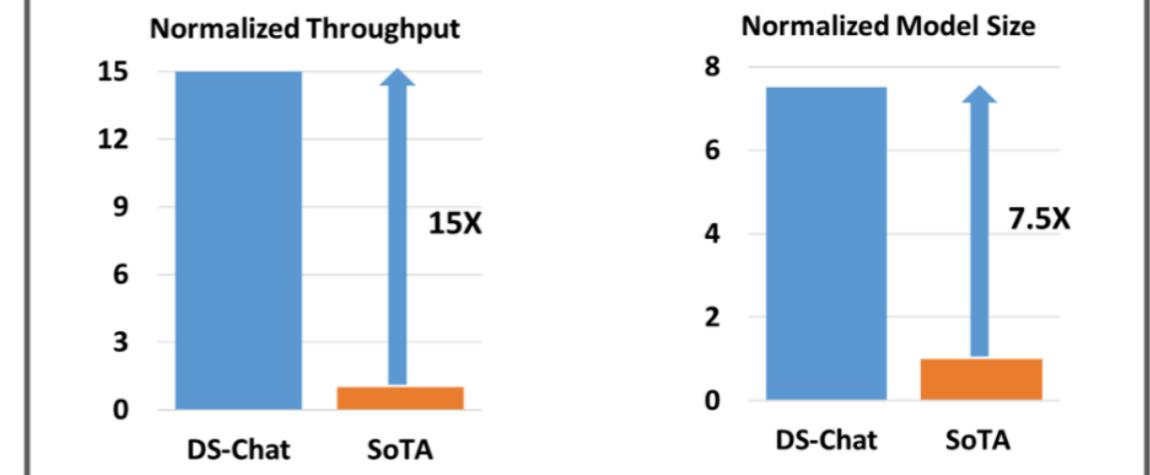


Microsoft Deepspeed Chat

- Enable training a baby “ChatGPT” w/ 4-8 A100 GPUs within 6-30hrs



Train 15X Faster and Scale to 5x Bigger Models than SOTA RLHFs



Easy-Breezy Training

A complete end-to-end RLHF training experience with a single click

High Performance System

Hybrid Engine achieves 15X training speedup over SOTA RLHF systems with unprecedented cost reduction at all scales

Accessible Large Model Support

Training ChatGPT-Style models with tens to hundreds of billions parameters on a single or multi-GPUs through ZeRO and LoRA

A Universal Acceleration Backend for RLHF

Support InstructGPT pipeline and large-model finetuning for various models and scenarios

Dialog Evaluation

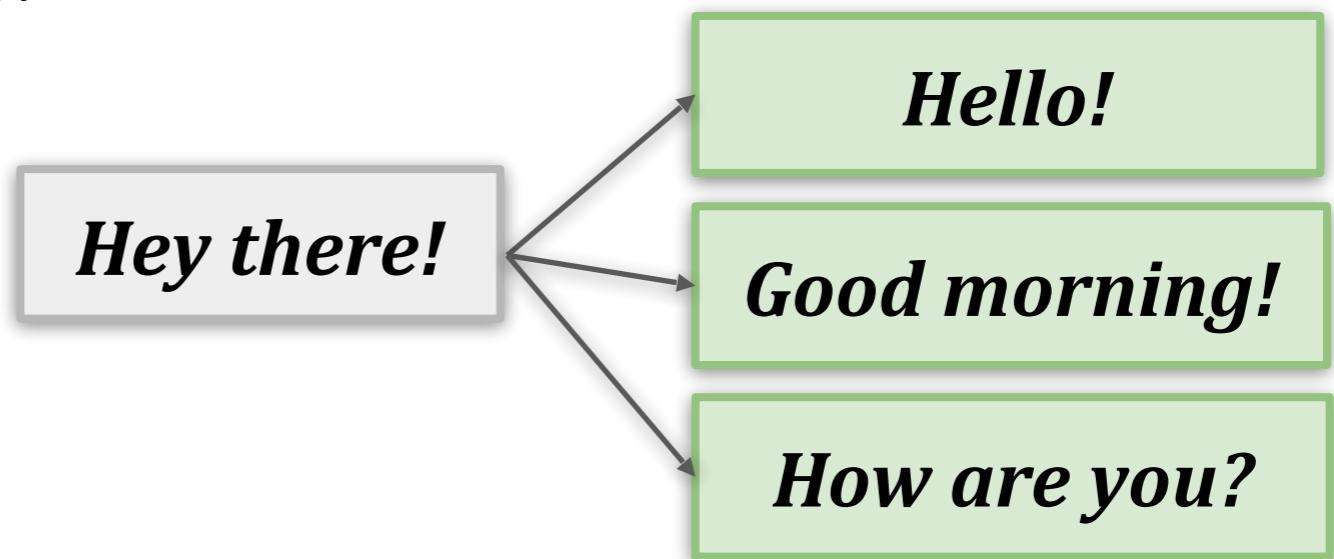
Dialog Evaluation

- **Goal:** Construct automatic evaluation metrics for response generation/interactive dialog
- **Given:** dialog history, generated response, reference response (optional)
- **Output:** a score for the response

Why is evaluating dialog hard? (1/3)

1. **One-to-many** nature of dialog

- For each dialog there are **many valid responses**
- Cannot compare to a reference response
 - The reference response isn't the only valid response
- Existing metrics won't work
 - BLEU, F-1, etc.



Why is evaluating dialog hard? (2/3)

2. Dialog quality is **multi-faceted**

- A response isn't just **good** or **bad**
- For interpretability, should measure **multiple qualities**
 - Relevance
 - Interestingness
 - Fluency

Why is evaluating dialog hard? (3/3)

3. Dialog is **inherently interactive**

- Dialog systems are designed to have a **back-and-forth interaction** with a user
 - Research largely focuses on **static corpora** → Reduces the problem of dialog to **response generation**
- Some properties of a system can't be assessed outside an interactive environment
 - Long-term planning, error recovery, coherence.

Dialog Evaluation

- Evaluation of dialog is **hard**
 - Can't compare to a reference response [no BLEU, F-1, etc.]
 - Should assess **many aspects** of dialog quality [relevant, interesting, etc.]
 - Should evaluate in an **interactive** manner

Dialog Evaluation

- USR [Mehri and Eskenazi. 2020]
- GRADE [Huang et al. 2020]
- HolisticEval [Pang et al. 2020]
- DSTC6 [Hori and Hori. 2017]
- FED [Mehri and Eskenazi. 2020]
- DSTC9 [Gunasekara et al. 2021]

<https://github.com/exe1023/DialEvalMetrics>

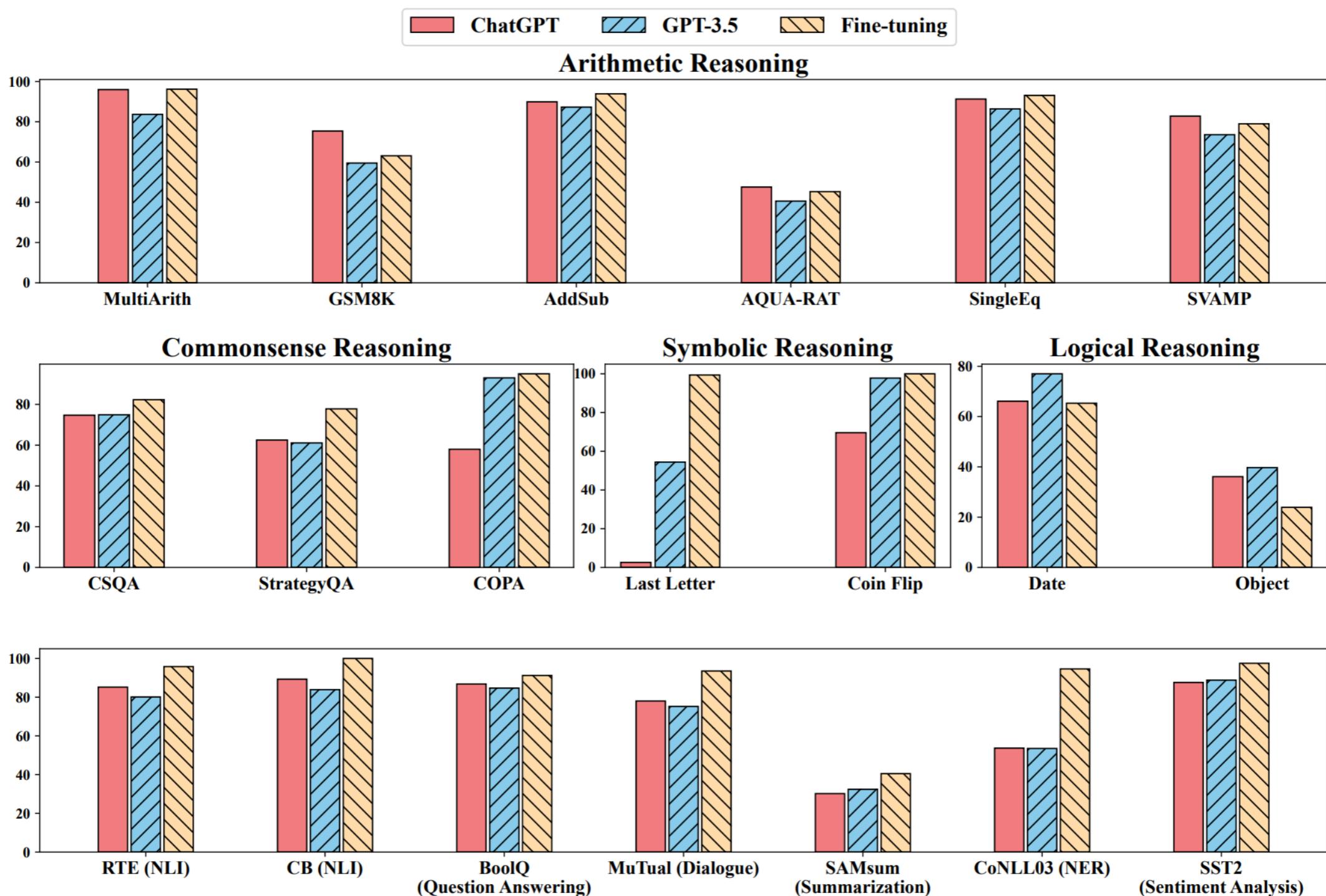
A Comprehensive Assessment of Dialog Evaluation Metrics

Yi-Ting Yeh, Maxine Eskenazi, Shikib Mehri

Evaluation of LLM Chatbot

- OpenAI released the evaluation benchmark: <https://github.com/openai/evals>
- Most of the responses look good to human, but get lower scores according to automatic evaluation metrics (BLEU, ROUGH-L, etc)
- Still an open question

Is ChatGPT a General-Purpose Natural Language Processing Task Solver? (Qin et al 2023)



Questions?