CS 224S / LINGUIST 285
Spoken Language Processing

Andrew Maas
Stanford University
Spring 2022

Lecture 6: End-to-end dialog approaches. Reinforcement learning in dialog

Original slides by Dan Jurafsky

Outline

- System architecture: Pipelined vs end-to-end (E2E)
- Deep learning in dialog components
- Reinforcement learning (RL) in end-to-end dialog systems
  - Quick introduction to RL
  - Recent research in end-to-end neural RL

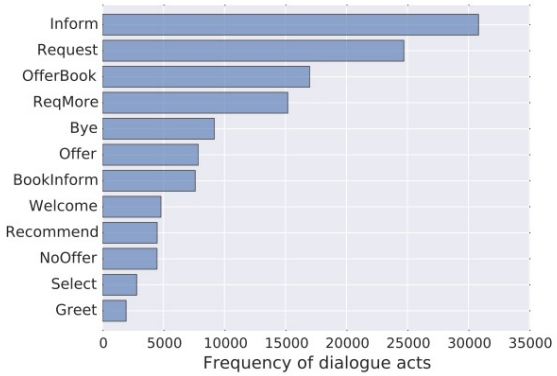
Learning to do dialog management

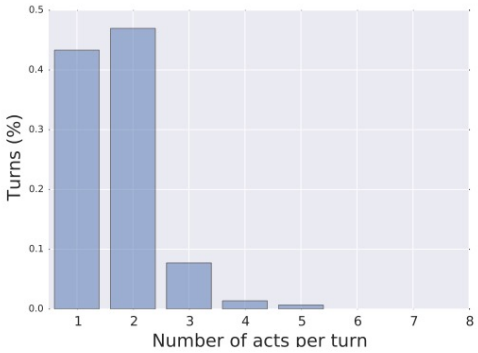
- Dialog state management via frames and rules
- As dialog complexity grows:
  - More possible actions/responses after each user utterance
  - Many choices about how dialog management *policy* completes tasks
- Hand engineering complex dialog policies is hard
- Can we *learn* dialog management? By optimizing:
  - Correct *response* given an *input*
  - Task completion and user satisfaction of the *full dialog*

Dialog Act Markup in Several Layers (DAMSL): forward looking function

STATEMENT	a claim made by the speaker
INFO-REQUEST	a question by the speaker
CHECK	a question for confirming information
INFLUENCE-ON-ADDRESSEE	(=Searle's directives)
OPEN-OPTION	a weak suggestion or listing of options
ACTION-DIRECTIVE	an actual command
INFLUENCE-ON-SPEAKER	(=Austin's commissives)
OFFER	speaker offers to do something
COMMIT	speaker is committed to doing something
CONVENTIONAL	other
OPENING	greetings
CLOSING	farewells
THANKING	thanking and responding to thanks

MultiWOZ Dataset and Dialog State Tracking





- You are traveling to Cambridge and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the type of **hotel** and should be in the **centre**.
- The hotel should **include free wifi** and should have **a star of 4**.
- Once you find the **hotel** you want to book it for **3 people** and **5 nights** starting from **monday**.
- Make sure you get the **reference number**.
- You are also looking for a **restaurant**. The restaurant should serve **australasian** food and should be in the **moderate** price range.
- The restaurant should be **in the same area as the hotel**.
- If there is no such restaurant, how about one that serves **british** food.
- Once you find the **restaurant** you want to book a table for **the same group of people** at **18:30** on **the same day**.
- Make sure you get the **reference number**

Figure 1: A sample task template spanning over three domains - hotels, restaurants and booking.

Table 2: Full ontology for all domains in our data-set. The upper script indicates which domains it belongs to. *: universal, 1: restaurant, 2: hotel, 3: attraction, 4: taxi, 5: train, 6: hospital, 7: police.

act type	inform* / request* / select[123] / recommend/[123] / not found[123] request booking info[123] / offer booking[1235] / inform booked[1235] / decline booking[1235] welcome* /greet* / bye* / reqmore*
slots	address* / postcode* / phone* / name[1234] / no of choices[1235] / area[123] / pricerange[123] / type[123] / internet[2] / parking[2] / stars[2] / open hours[3] / departure[45] destination[45] / leave after[45] / arrive by[45] / no of people[1235] / reference no.[1235] / trainID[5] / ticket price[5] / travel time[5] / department[7] / day[1235] / no of days[123]

MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling

Paweł Budzianowski[1], Tsung-Hsien Wen[2*], Bo-Hsiang Tseng[1], Iñigo Casanueva[2*], Stefan Ultes[1], Osman Ramadan[1] and Milica Gašić[1]
[1]Department of Engineering, University of Cambridge, UK,
[2]PolyAI, London, UK
{pfb30,mg436}@cam.ac.uk

MultiWOZ Example

Table 2: An example of partial dialog for MultiWOZ 2.1 dataset.

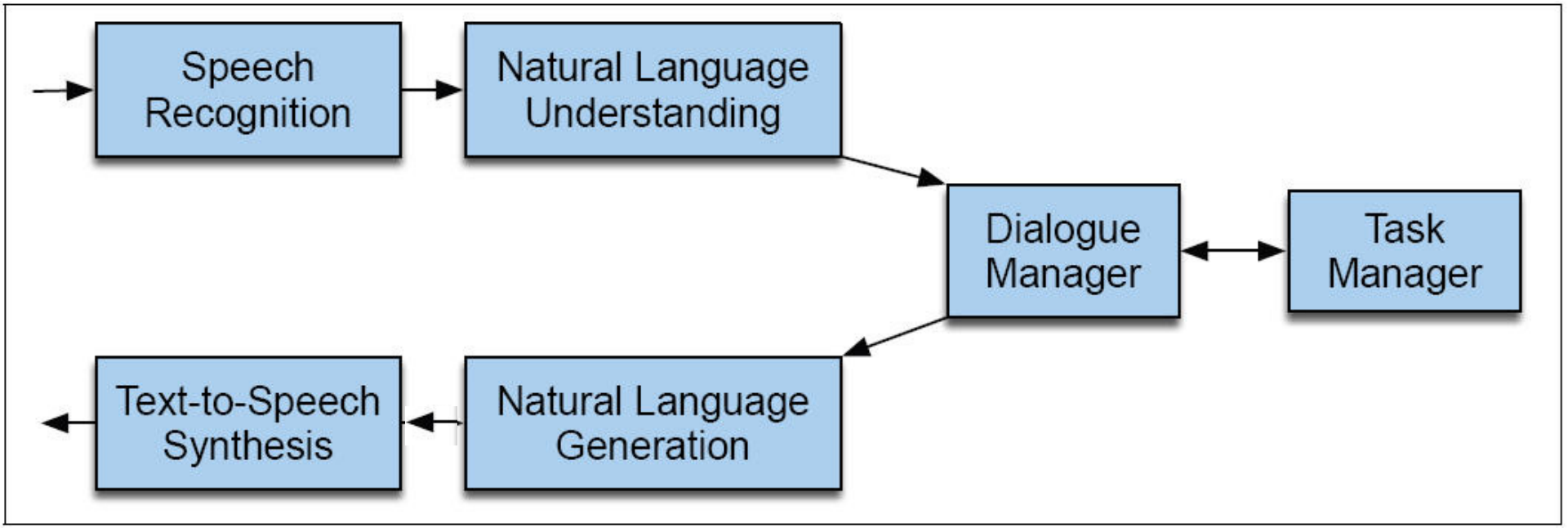
Speaker	Utterance	Dialog States
User	I am leaving Cambridge . I need a train that leaves after 13:45 . 我要离开剑桥。我需要一列在13:45之后离开的火车。	
System	On what day do you wish to travel ? 您想在哪一天旅行?	train: {leaveAt: 13:45, departure: cambridge } 火车: {出发时间: 13:45, 出发地: 剑桥}
User	I am traveling on Wednesday and need to go to Birmingham New Street please . 我打算星期三旅行，我需要去伯明翰新街。	
System	TR1766 . It is going to take 163 minutes . can i book it for you ? 您可以乘坐TR1766。这将需要163分钟。需要我为您预订吗?	train: { destination: birmingham new street, leaveAt: 13:45, departure: cambridge, day: wednesday } 火车: {目的地: 伯明翰新街, 出发时间: 13:45, 出发地: 剑桥,天: 星期三}
User	Yes , can you book 4 tickets for me ? 好的，你能帮我预订四张票吗?	
System	I have you booked on TR1766 leaving Cambridge at 18:01 arriving Birmingham New Street at 20:44 . Your reference number is 6L0MANZV . Can I help with anything else ? 我帮您订了TR1766的票，它于18:01离开剑桥，于20:44到达伯明翰新街。您的参考号码是6L0MANZV。您还需要别的帮助吗?	train: { destination: birmingham new street, leaveAt: 13:45, departure: cambridge, day: wednesday, book: {people: 4, booked: [trainID: TR1766, reference: 6L0MANZV] } } 火车: {目的地: 伯明翰新街, 出发时间: 13:45, 出发地: 剑桥, 天: 星期三, 预订: {人数: 4, 预订成功: [火车号: TR1766, 参考号: 6L0MANZV] } },
User	Also looking for a place that has British food and moderately priced . 我也正在寻找一个有英国美食且价格适中的地方。	
System	I have many options for you . They are all in the city center . Do you have any other preferences ? 我有很多选择给你。它们都在市中心。您还有其他偏好吗?	train: { destination: birmingham new street, leaveAt: 13:45, departure: cambridge, day: wednesday, book: {people: 4, booked: [trainID: TR1766, reference: 6L0MANZV] } } 火车: {目的地: 伯明翰新街,出发时间: 13:45, 出发地: 剑桥, 天: 星期三, 预订: {人数: 4, 预订成功: [火车号: TR1766,参考号: 6L0MANZV]} }

See: Dialogue System Technology Challenge

Dialog management in complex systems

- Representing state/actions for dialog management gets complex fast for systems beyond just form-filling
- Need to:
  - Decide when the user has asked a question, made a proposal, rejected a suggestion
  - Ground a user's utterance, ask clarification questions, suggest plans
- Need to design representation and transitions
  - Model user mental state? More than just list of slots
  - Many possible responses at any turn (clarification? Move to new subtopic? Confirm/ground?)

Pipelined Conceptual Architecture so far

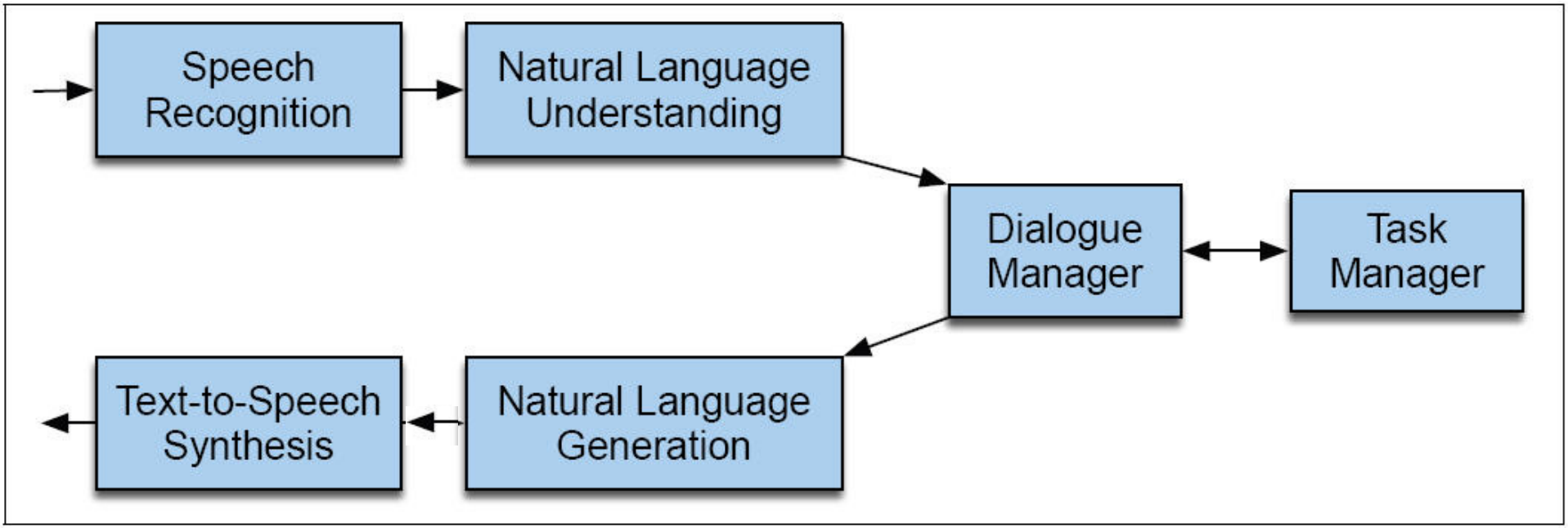


Pipelined Conceptual Architecture pros/cons

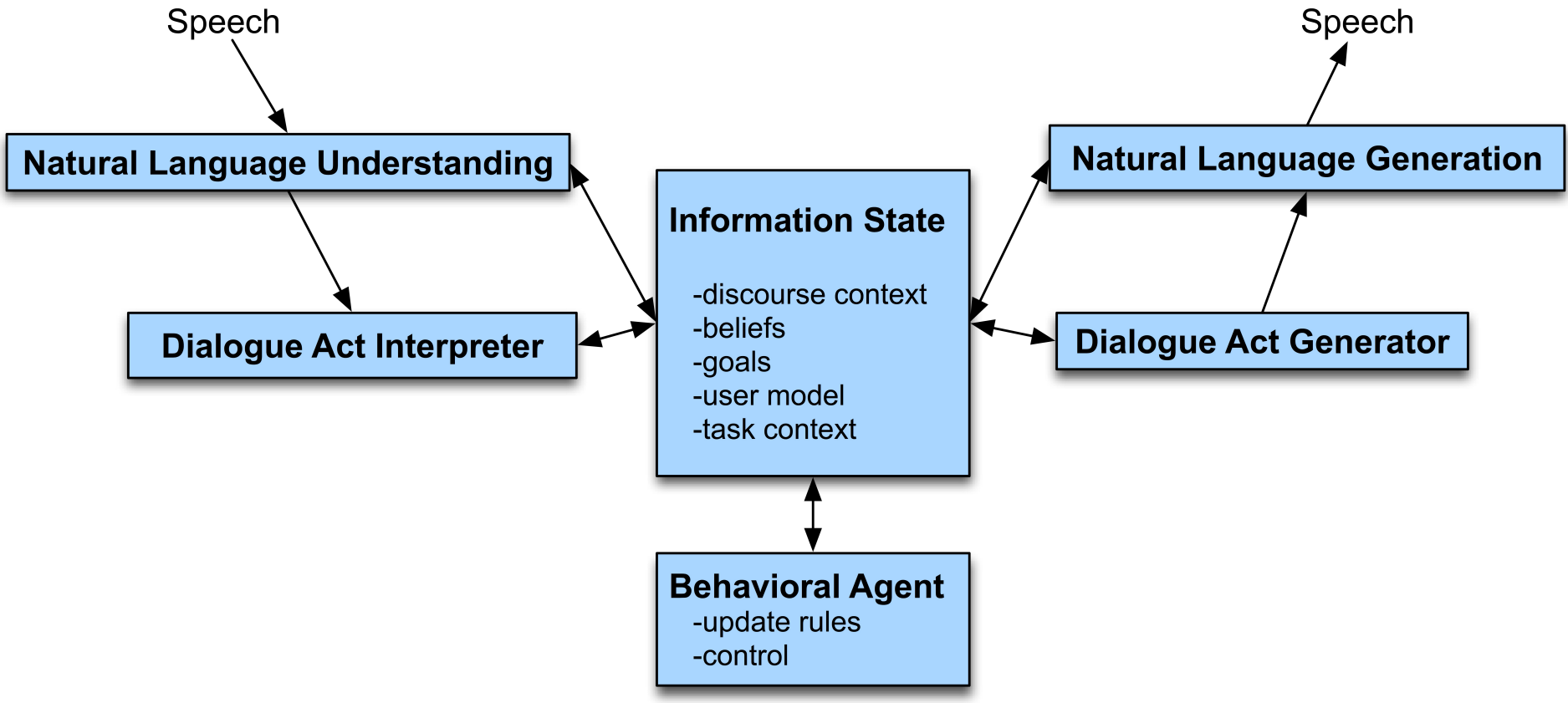
- + Clear component boundaries.
- + Train/eval each in isolation
  - Different datasets can help (e.g. large corpus for ASR)
  - Reuse high quality components when possible (ASR/TTS)
  - Per-component metrics and labels
- + Direct control over each component/interface
- - Hand engineering dialog representations is hard!
- - Interfaces force decisions about what information to pass between components. Often sub-optimal
- - True metric is task completion / satisfaction. Unclear how optimizing component metrics affect overall performance

End-to-end Architectures

- What if we could learn an *end-to-end* system that directly generates actions and responses given user input?
  - + Eliminate hand engineered interfaces between components that limit information
  - + Optimize entire system for end goal vs per-component metrics/labels
  - - Potentially hard to debug, and end-to-end training might limit component/data transfer
- Especially important for multi-domain or personal assistant systems where many states/actions possible


Speech Recognition
Natural Language Understanding
Dialogue Manager
Task Manager
Text-to-Speech Synthesis
Natural Language Generation

Information-state



Dialog state tracking as a task

- Predict state (slot values + dialog act) from

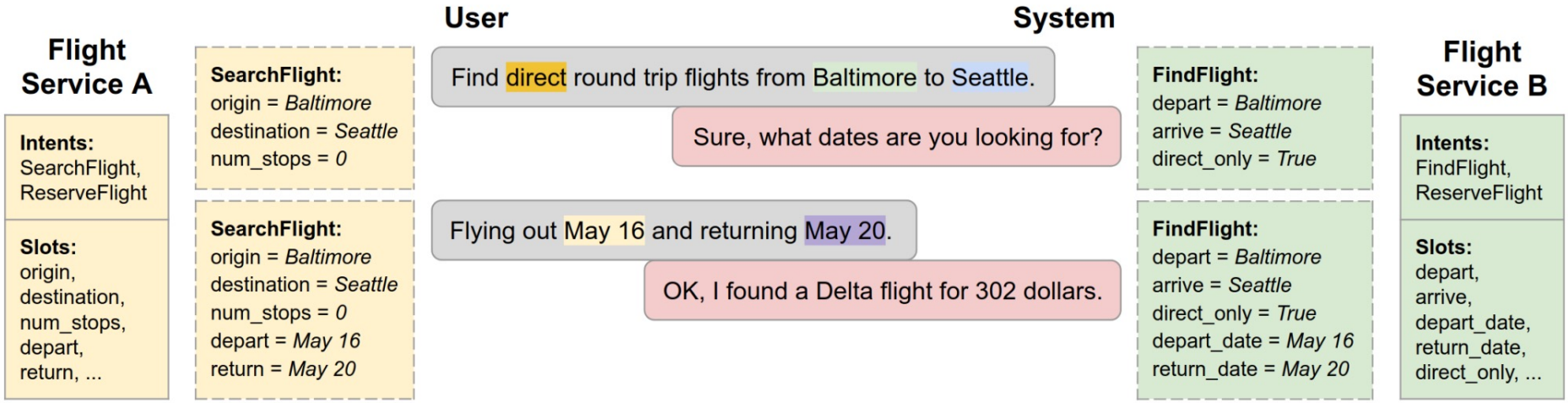


Figure 2: Dialogue state tracking labels after each user utterance in a dialogue in the context of two different flight services. Under the schema-guided approach, the annotations are conditioned on the schema (extreme left/right) of the underlying service.

DSTC8 overview

Dialog state tracking

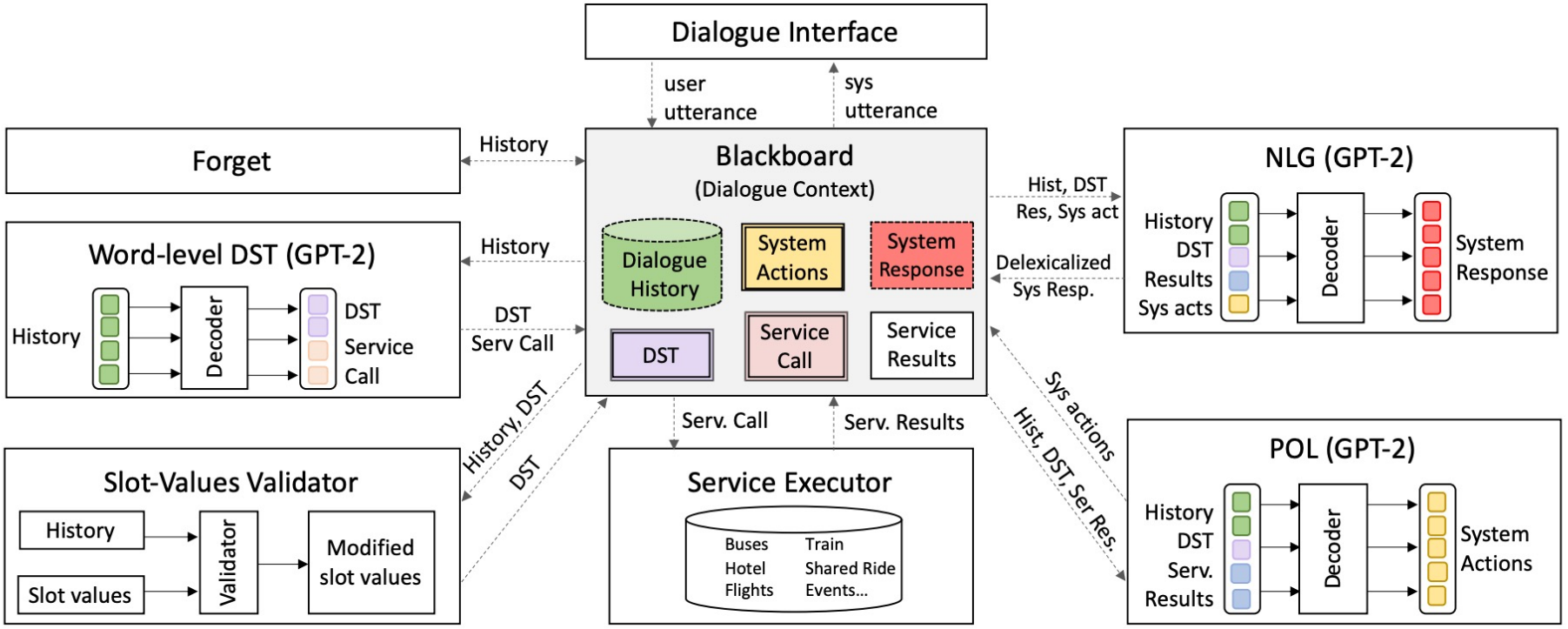


Figure 1: Dialogue System Architecture. Arrows illustrate retrieving/updating information from/to the blackboard. Boxes labeled *GPT-2* (DST, POL, NLG) represent the same neural module which is invoked multiple times using different aggregated inputs. Dotted boxes contain symbols and double-line boxes contain natural language.

(Romero *et al*, 2021

MultiWOZ Single domain example

Dialogue id : "SNG0689"

Goal

Domain : "restaurant"

Informable	"food" : "british" "area" : "west"
Requestable	"phone" "address" "postcode"
Book	"people" : 6

Database (restaurant)

"ref" : "00000081"
"name" : "graffiti"
"phone" : "01223277977"
"postcode" : "cb30lx"
"address" : "Hotel Felix Whitehouse Lane Hunthindon Road"
"food" : "british"
"area" : "west"
⋮

"ref" : "00000084"
"name" : grafton hotel restaurant
"phone" : "01223241387"
"postcode" : "cb580a"
"address" : "Grafton Hotel 619 Newmarket Road Fen Ditton"
"food" : "british"
"area" : "east"
⋮

Dialogue turns

[usr] Are there any restaurants that serve proper **British** food in town?
[sys] Oh yes quite a few. Which part of town will you be dining in?
[usr] **West**, if possible.
[sys] We have three: Graffiti, Saint John's Chop House, and Traveller.
[usr] Will you give me the **phone number**, **address**, and **postcode** for Graffiti?
[sys] The phone number is **01223277977**. The address is **Hotel Felix Whitehouse Lane Hunthindon Road** and post code is **cb30lx**. Want to book?
[usr] 6 people, please.
[sys] Booking is successful. Reference number is **00000081**.

Blue : Informable slot **Yello-Green : Requestable slot name** **Orange : Requestable slot value**

(Ham et al. ACL 2020)

End-to-end neural pipeline for goal-oriented dialogue systems using GPT2

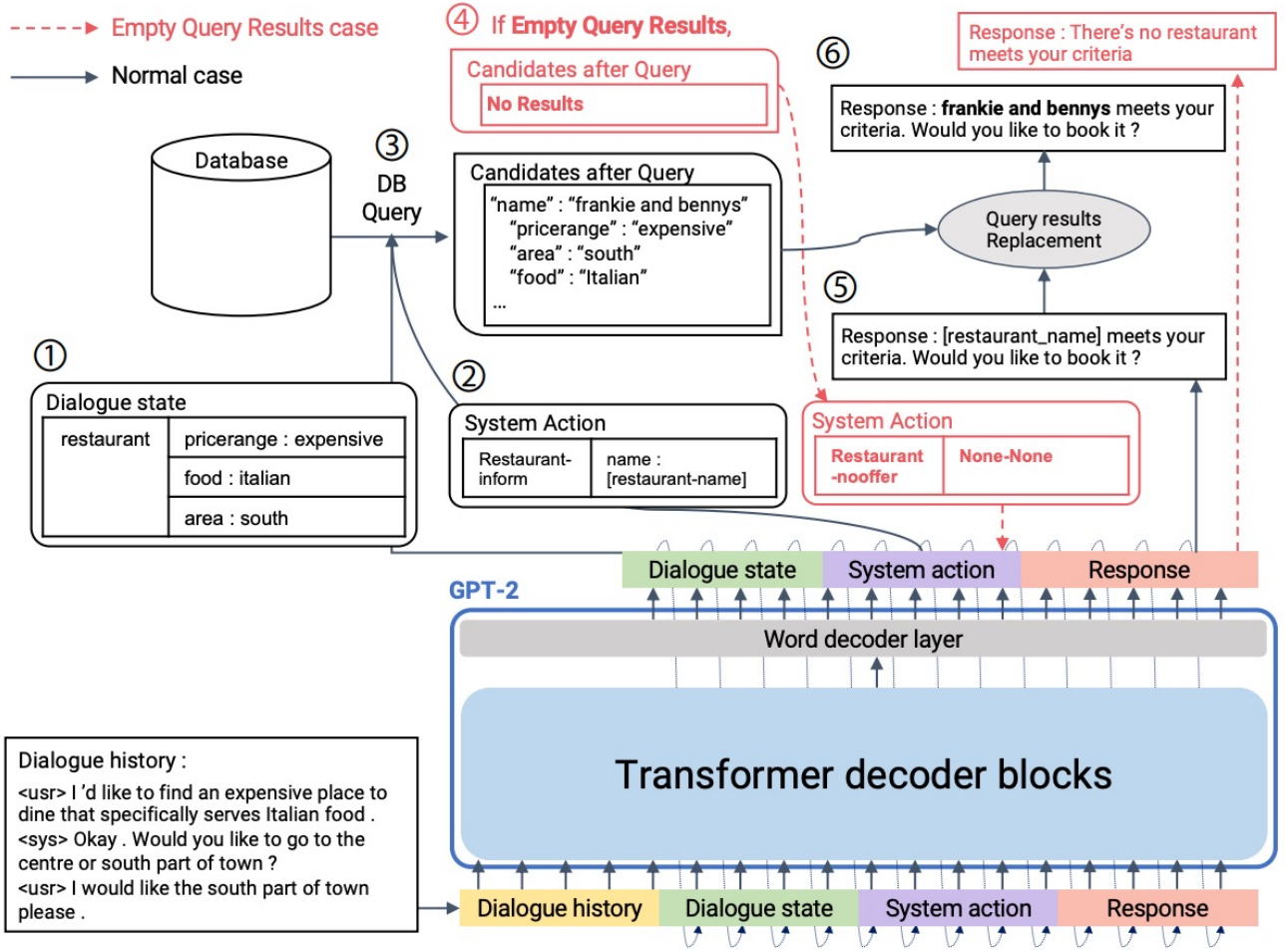


Figure 2: The overview of our end-to-end neural dialogue model. For the transformer, we use fine-tuned GPT-2. The dashed line represents the information to and from the DB query, which is invoked when the system action needs to fetch an actual value from the database.

(Ham et al. ACL 2020)

End-to-end neural pipeline for goal-oriented dialogue systems using GPT2

- Key insights for other E2E neural approaches::

- Output each module-specific inference using shared encoder
- Fine-tune encoder from GPT-2 →Transfer learning

(Ham et al. ACL 2020)

End-to-end neural pipeline for goal-oriented dialogue systems using GPT2

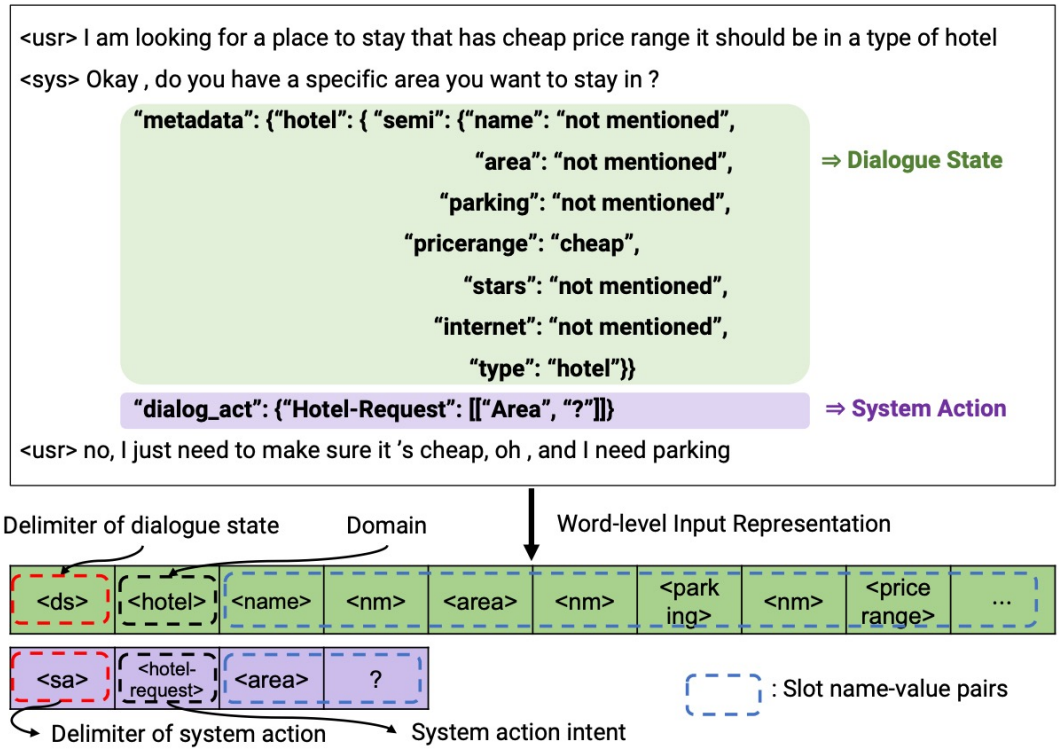


Figure 3: In the MultiWOZ dataset, the 'metadata' is treated as the dialogue state and the 'dialogue act' is treated as the system action.

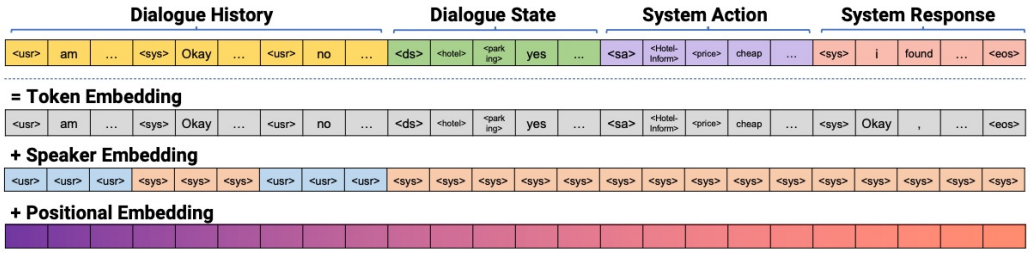


Figure 4: Input representation for fine-tuning GPT-2.

(Ham et al. ACL 2020)

End-to-end neural pipeline for goal-oriented dialogue systems using GPT2

Rank	Team ID	Success Rate ↑	Language Understanding ↑	Response Appropriateness ↑	Turns ↓
1	**OURS**(504430)	**68.32%**	**4.149**	**4.287**	19.507
2	504429	65.81%	3.538	3.632	15.481
3	504563	65.09%	3.538	3.840	**13.884**
4	504651	64.10%	3.547	3.829	16.906
5	504641	62.91%	3.742	3.815	14.968
N/A	Baseline	56.45%	3.097	3.556	17.543

Table 2: Overall results of the human evaluation carried out by DSTC8 organizers. Only the top five teams and the baseline results are compared.

Model	Joint Acc.	Slot Acc.
GLAD (Zhong et al., 2018)	35.57	95.44
GCE (Nouri and Hosseini-Asl, 2018)	36.27	98.42
SUMBT (Lee et al., 2019a)	46.64	96.44
TRADE (Wu et al., 2019)	**48.62**	**96.92**
OURS + greedy	44.03	96.07

Table 3: Performance comparison with other state-of-the-art models in Dialogue State Tracking benchmark of MultiWOZ dataset.

Model	Inform	Success	BLEU
BASELINE (Budzianowski et al., 2018)	71.29	60.96	18.80
TOKENMOE (Pei et al., 2019)	75.30	59.70	16.81
HDSA (Chen et al., 2019)	82.9	68.90	**23.60**
STRUCTURED FUSION (Mehri et al., 2019)	82.70	72.10	16.34
LARL (Zhao et al., 2019)	**82.78**	**79.20**	12.80
OURS + greedy	77.00	69.20	6.01

Table 4: Performance comparison with other state-of-the-art models in Dialogue-Context-to-Text Generation benchmark of MultiWOZ dataset.

(Ham et al. ACL 2020)

Neural end-to-end trainable task-oriented dialogue systems

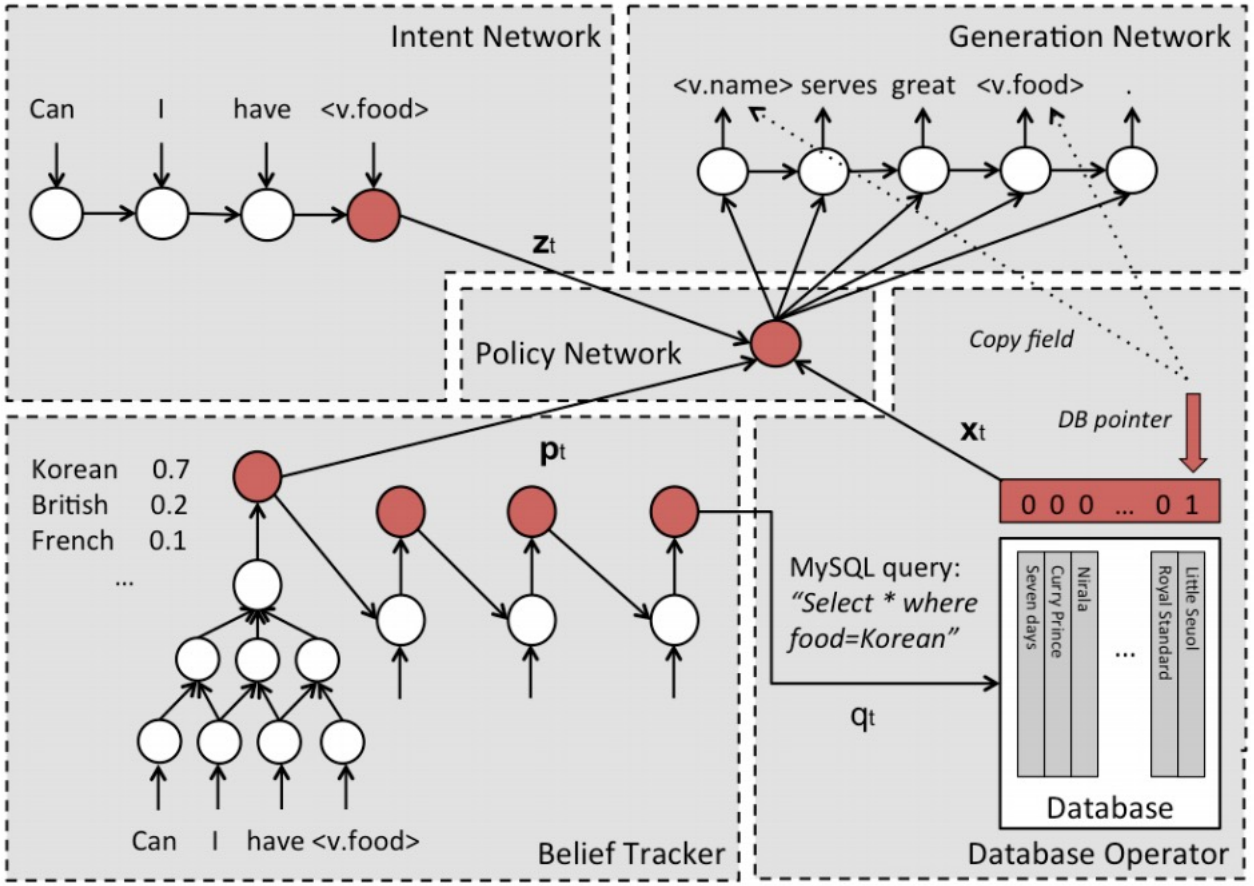


Figure 1: The proposed end-to-end trainable dialogue system framework

(Wen et al. EACL 2017)

Modeling a dialogue system as a probabilistic agent

- The current knowledge of the system
  - Set of states S the agent can be in
- Set of actions A the agent can take
- A success metric that tells us how well the agent achieved its goal
- A way of using this metric to learn a policy π for what action to take in any state.
  (Reinforcement Learning)

What do we mean by actions A and policies π?

- Kinds of decisions a conversational agent needs to make:
  - When should I ground/confirm/reject/ask for clarification on what the user just said?
  - When should I ask a directive prompt, when an open prompt?
  - When should I use user, system, or mixed initiative?

Markov Decision Processes

- Or MDP
- Characterized by:
  - a set of states S an agent can be in
  - a set of actions A the agent can take
  - A reward r(a,s) that the agent receives for taking an action in a state
- Learn from human-human examples, or learn online by interactions
- More thorough RL intros: CS234 videos online. Berkeley CS285

What is a state?

- In principle, MDP state could include any possible information about dialogue
- Usually use a much more limited set due to compute limits
  - Values of slots in current frame
  - Most recent question asked to user
  - User's most recent answer
  - ASR confidence, *etc.*
- In deep learning approaches, state might be implicit. NN predicts next action from NLU input + history/memory neural representations

MDP

- We can think of a dialogue as a trajectory in state space

$$s_1 \xrightarrow{a1,r1} s_2 \xrightarrow{a2,r2} s_3 \xrightarrow{a3,r3} \cdots$$

- The best policy π^* is the one with the greatest expected reward over all trajectories
- How to compute a reward for a state sequence?

Reward for a state sequence

- Central RL theme: discounted rewards
- Cumulative reward Q of a sequence is discounted sum of utilities of individual states

$$Q([s_0, a_0, s_1, a_1, s_2, a_2 \cdots]) = R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots,$$

- Discount factor γ between 0 and 1
- Makes agent care more about current than future rewards; the more future a reward, the more discounted its value

The Markov assumption

- MDP assumes that state transitions are Markovian

$$P(s_{t+1} \mid s_t, s_{t-1}, \ldots, s_o, a_t, a_{t-1}, \ldots, a_o) = P_T(s_{t+1} \mid s_t, a_t)$$

Expected reward for an action

- Expected cumulative reward Q(s,a) for taking a particular action from a particular state can be computed by Bellman equation:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

- Expected cumulative reward for a given state/action pair is:
  - immediate reward for current state
  - + expected discounted utility of all possible next states s'
  - Weighted by probability of moving to that state s'
  - And assuming once there we take optimal action a'

What we need for Bellman equation

- A model of p(s'|s,a)
- Estimate of R(s,a)

How to get these?

- If we had labeled training data
  - P(s'|s,a) = C(s,s',a)/C(s,a)
- If we knew the final reward for whole dialogue $R(s_1,a_1,s_2,a_2,\ldots,s_n)$
- Given these parameters, can use value iteration algorithm to learn Q values (pushing back reward values over state sequences) and hence best policy

How to estimate p(s'|s,a) without labeled data

Have random conversations with real people:

- Carefully hand-tune small number of states and policies
- Then can build a dialogue system which explores state space by generating a few hundred random conversations with real humans
- Set probabilities from this corpus

Have random conversations with simulated people:

- Now you can have millions of conversations with simulated people
- Allows for larger state space

Neural dialog learning with human teaching and RL

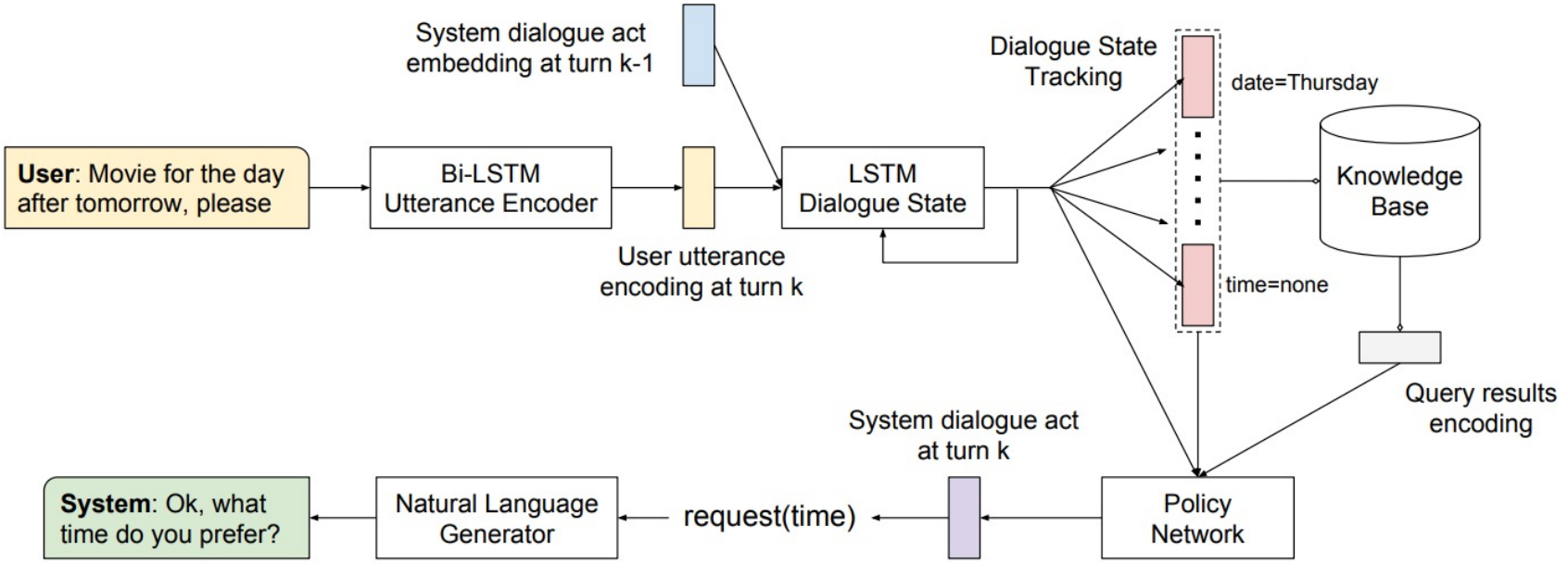


Figure 1: Proposed end-to-end task-oriented dialogue system architecture.

(Liu et al. ACL 2020)

Neural dialog learning with human teaching and RL

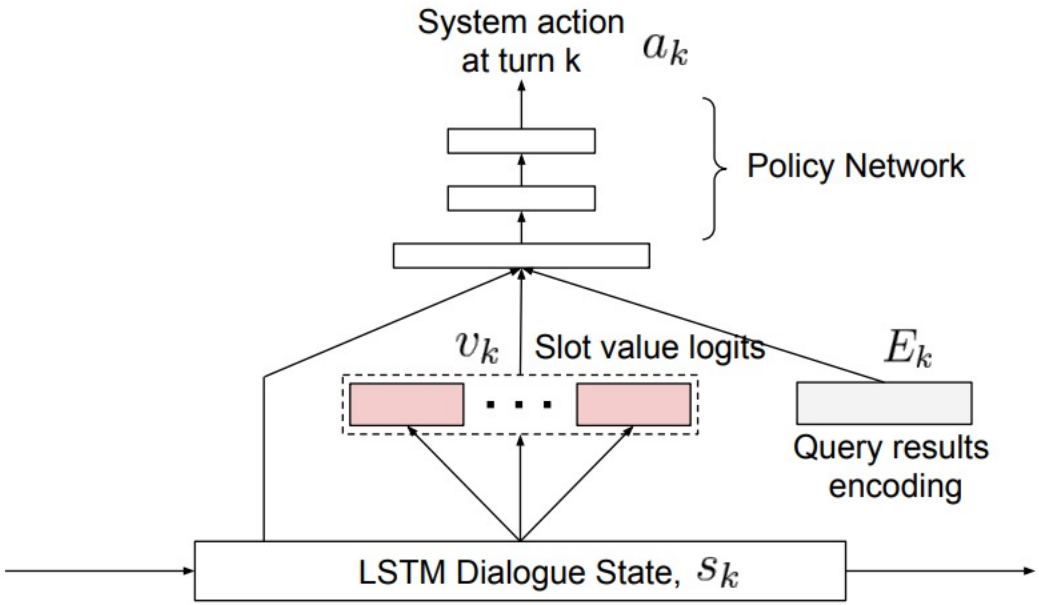


Figure 2: Dialogue state and policy network.

$$P(a_k \mid U_{\leq k},\ A_{<k},\ E_{\leq k}) = \text{PolicyNet}(s_k, v_k, E_k) \tag{3}$$

(Liu et al. ACL 2020)

Neural dialog learning with human teaching and RL

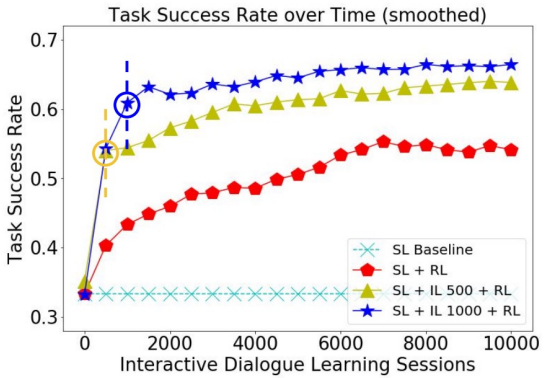


Figure 3: Interactive learning curves on task success rate.

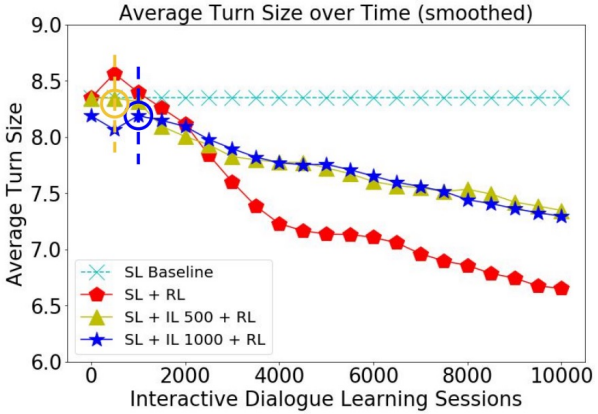


Figure 4: Interactive learning curves on average dialogue turn size.

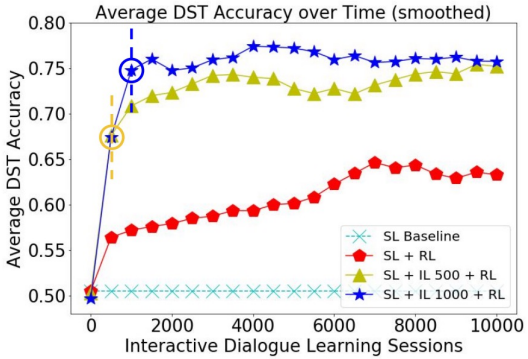


Figure 5: Interactive learning curves on dialogue state tracking accuracy.

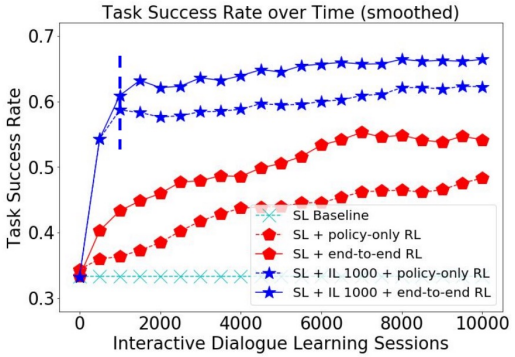


Figure 6: Interactive learning curves on task success rate with different RL training settings.

(Liu et al. ACL 2020)

E2E LSTM-based dialog control with RL

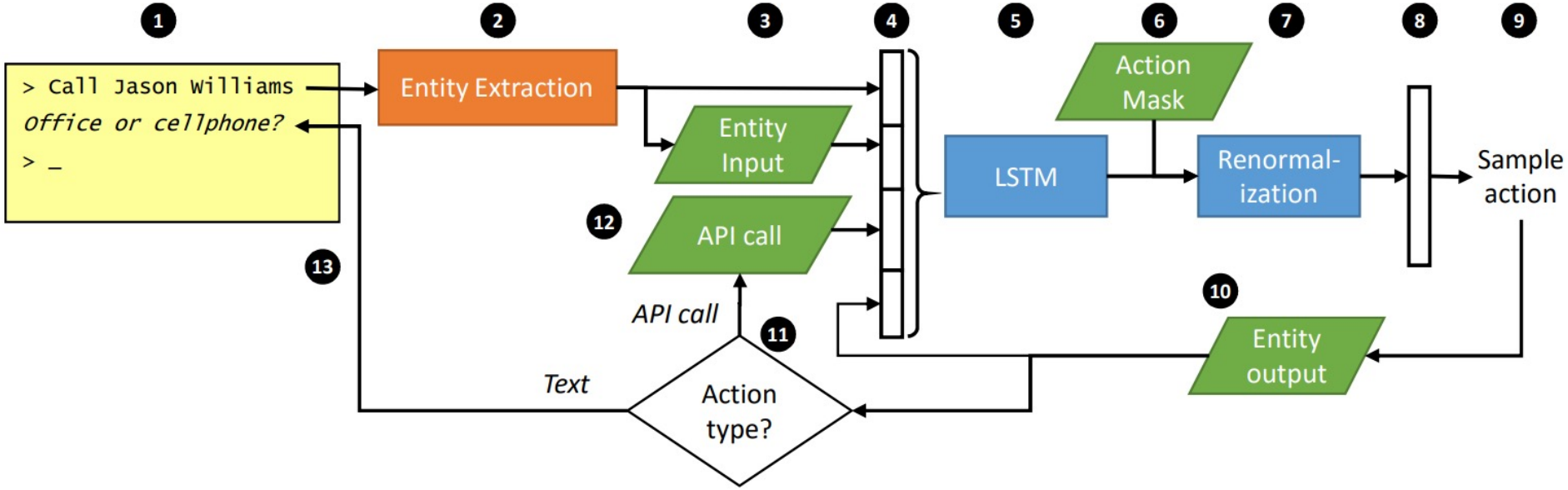
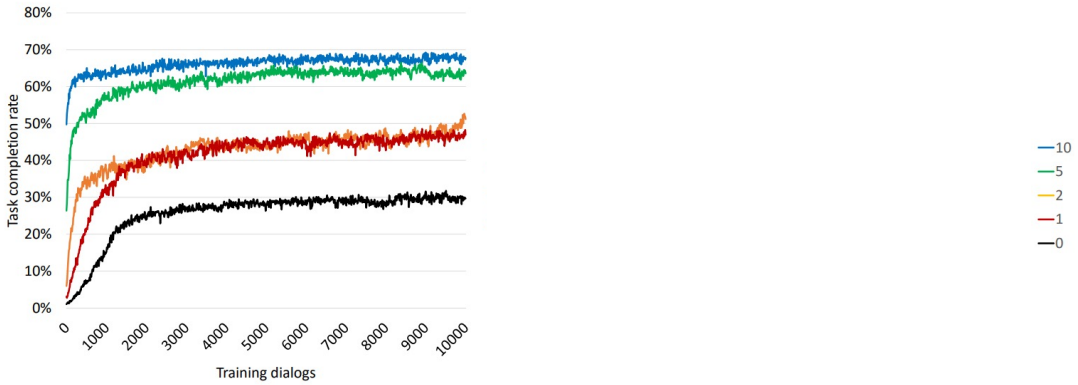


Figure 1: Operational loop. Green trapezoids refer to programmatic code provided by the software developer. The blue boxes indicate the recurrent neural network, with trainable parameters. The orange box performs entity extraction. The vertical bars in steps 4 and 8 are a feature vector and a distribution over template actions, respectively. See text for a complete description.

(Williams & Zweig. 2017)

E2E LSTM-based dialog control with RL



(a) TCR mean.

Figure 5: Task completion rate (TCR) mean and standard deviation for a policy initially trained with $N = (0, 1, 2, 5, 10)$ dialogs using supervised learning (SL), and then optimized with 0 to $10,000$ dialogs using reinforcement learning (RL). Training and evaluation were done with the same stochastic simulated user. Each line shows the average of 10 runs, where the dialogs used in the SL training in each run were randomly sampled from the 21 example dialogs.

How can I help you?
Call Jason
Which type of phone: mobile or work?
Oh, actually call Mike on his office phone
Calling Michael Seltzer, work.
`PlaceCall`

How can I help you?
Call Jason at his home
Sorry, I don't have a home number for Jason Williams. I have mobile or work. Which would you like?
mobile
Calling Jason Williams, mobile
`PlaceCall`

How can I help you?
Call Michel
Sorry, I don't know of any names called Michel. Can you try again?
Call Michael Seltzer
`SavePhonetypeavail`
Calling Michael Seltzer, work
`PlaceCall`

(Williams & Zweig. 2017)

State of the art

1. Use deep learning approaches for ASR/TTS. Keep these modules separate
2. Build shared encoder with task outputs for intermediate state (NLU slots, actions, belief state, DB query info)
3. Training options (depends on availability):
    1. Initialize encoder with pre-trained weights (e.g. GPT2)
    2. Use supervised learning to optimize per-task outputs
    3. Use supervised/imitation learning to optimize end-to-end (match desired output/action and backprop all the way back)
    4. Interact with live/simulated users for reinforcement learning

Appendix

An example of dialogue act detection: Correction Detection

- If system misrecognizes an utterance, and either
  - Rejects
  - Via confirmation, displays its misunderstanding
- Then user has a chance to make a

  correction
  - Repeat themselves
  - Rephrasing
  - Saying “no” to the confirmation question.

Corrections

- Unfortunately, corrections are harder to recognize than normal sentences!
  - Swerts et al (2000): corrections misrecognized twice as often (in terms of WER) as non-corrections!!!
  - Why?
    - Prosody seems to be largest factor: **hyperarticulation**
    - Liz Shriberg example:
      - "NO, I am DE-PAR-TING from Jacksonville"

DAMSL: backward looking function

AGREEMENT speaker's response to previous proposal
- ACCEPT accepting the proposal
- ACCEPT-PART accepting some part of the proposal
- MAYBE neither accepting nor rejecting the proposal
- REJECT-PART rejecting some part of the proposal
- REJECT rejecting the proposal
- HOLD putting off response, usually via subdialogue

ANSWER answering a question

UNDERSTANDING whether speaker understood previous
- SIGNAL-NON-UNDER. speaker didn't understand
- SIGNAL-UNDER. speaker did understand
  - ACK demonstrated via continuer or assessment
  - REPEAT-REPHRASE demonstrated via repetition or reformulation
  - COMPLETION demonstrated via collaborative completion

A DAMSL Labeling

[info-req,ack]	A_1:	And, what day in May did you want to travel?
[assert, answer]	C_2:	OK uh I need to be there for a meeting that's from the 12th to the 15th.
[info-req,ack]	A_2:	And you're flying into what city?
[assert,answer]	C_3:	Seattle.
[info-req,ack]	A_3:	And what time would you like to leave Pittsburgh?
[check,hold]	C_4:	Uh hmm I don't think there's many options for non-stop.
[accept,ack]	A_4:	Right.
[assert]		There's three non-stops today.
[info-req]	C_5:	What are they?
[assert, open-option]	A_5:	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.
[accept,ack]	C_6:	OK I'll take the 5ish flight on the night before on the 11th.
[check,ack]	A_6:	On the 11th?
[assert,ack]		OK. Departing at 5:55pm arrives Seattle at 8pm, U.S. Air flight 115.

Older example

Singh, S., D. Litman, M. Kearns, and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of AI Research.

- NJFun system, people asked questions about recreational activities in New Jersey
- Idea of paper: use reinforcement learning to make a small set of optimal policy decisions

Very small # of states and acts

- States: specified by values of 8 features
  - Which slot in frame is being worked on (1-4)
  - ASR confidence value (0-5)
  - How many times a current slot question had been asked
  - Restrictive vs. non-restrictive grammar
  - Result: 62 states
- Actions: each state only 2 possible actions
  - Asking questions: System versus user initiative
  - Receiving answers: explicit versus no confirmation.

Ran system with real users

- 311 conversations
- Simple binary reward function
  - 1 if competed task (finding museums, theater, winetasting in NJ area)
  - 0 if not
- System learned good dialogue strategy: Roughly
  - Start with user initiative
  - Backoff to mixed or system initiative when re-asking for an attribute
  - Confirm only a lower confidence values