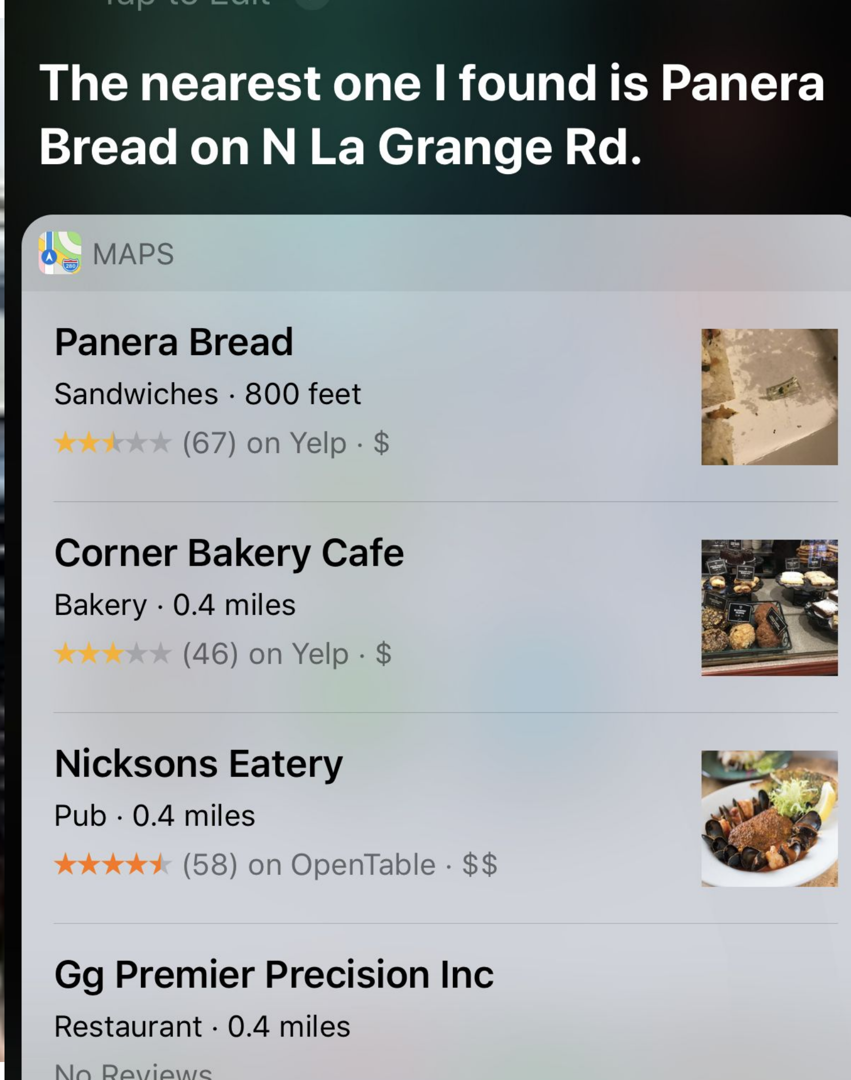
Deep Generation
in Task-Oriented Dialogue System

Lei Shu

https://leishu02.github.io/

Roadmap

- **Introduction to Task-Oriented Dialogue System**
- Modeling Multi-Action for Task-Oriented Dialogues
  Shu et al. *EMNLP 2019*
- Flexible Structured Model for Task-Oriented Dialogues
  Shu et al. *SIGDIAL 2019, NeurIPS 2018 Conversational AI Workshop*

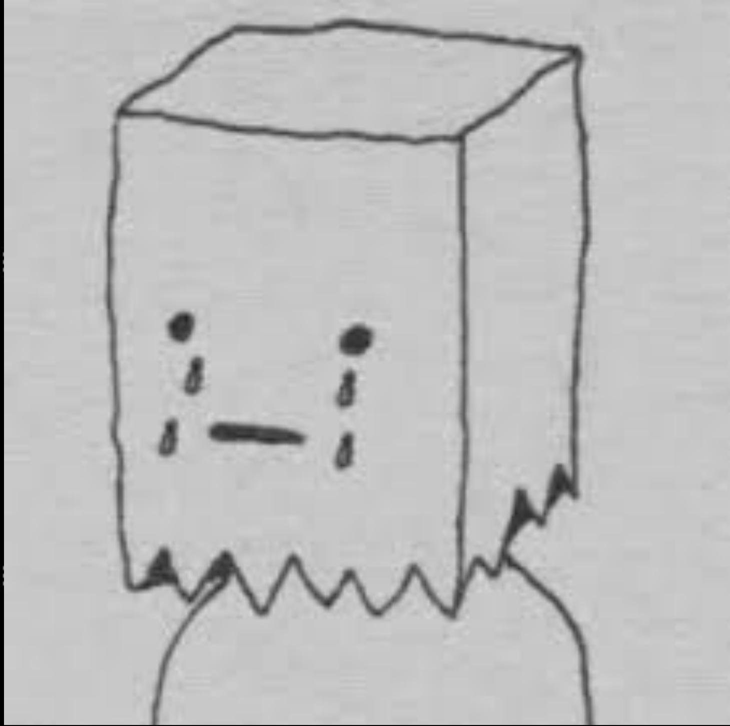


The nearest one I found is Panera Bread on N La Grange Rd.
MAPS
Panera Bread
Sandwiches · 800 feet
(67) on Yelp · $
Corner Bakery Cafe
Bakery · 0.4 miles
(46) on Yelp · $
Nicksons Eatery
Pub · 0.4 miles
(58) on OpenTable · $$
Gg Premier Precision Inc
Restaurant · 0.4 miles
No Reviews

Hey, Google, turn off
the kitchen (light)!

```
==============================================================================
          EEEEEEEE  L         IIIIIII  ZZZZZZZ      AAA
          E         L            I          Z      A   A
          E         L            I         Z      A     A
          EEEEE     L            I        Z       A     A
          E         L            I       Z        AAAAAAA
          E         L            I      Z         A     A
          EEEEEEEE  LLLLLLLL  IIIIIII  ZZZZZZ     A     A
==============================================================================
ELIZA > (PLEASE TYPE IN ALL CAPS) WHAT'S YOUR NAME DEAR ?
AMIT
ELIZA > HI, I'M ELIZA. WHAT DO YOU WANT TO TALK ABOUT ?
AMIT > CAN I TALK ABOUT MY PROBLEM ?
ELIZA > SURE... !
AMIT > DUE TO ON GOING EXAMS IN COLLEGE, I HAVE A LOT OF STRESS.
ELIZA > PLEASE RELAX A LITTLE. SLEEP WELL.
AMIT > THANKS FOR YOUR ADVICE.
ELIZA > NO MENTION.
AMIT > BYE
ELIZA > BYE AND KEEP IN TOUCH...

==============================================================================
```



Task-Oriented

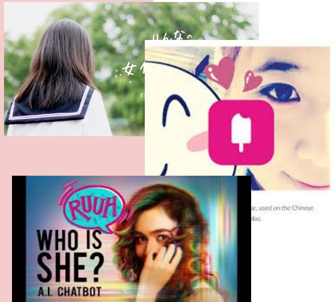
- ***"I need to get this done"***
- narrow focus
- goal-oriented, task-completion
  eg: buying movie ticket,
  in-car assistant
- efficient as possible
- understanding user's intention, tracking the dialogue history and finding next action

Chit Chat

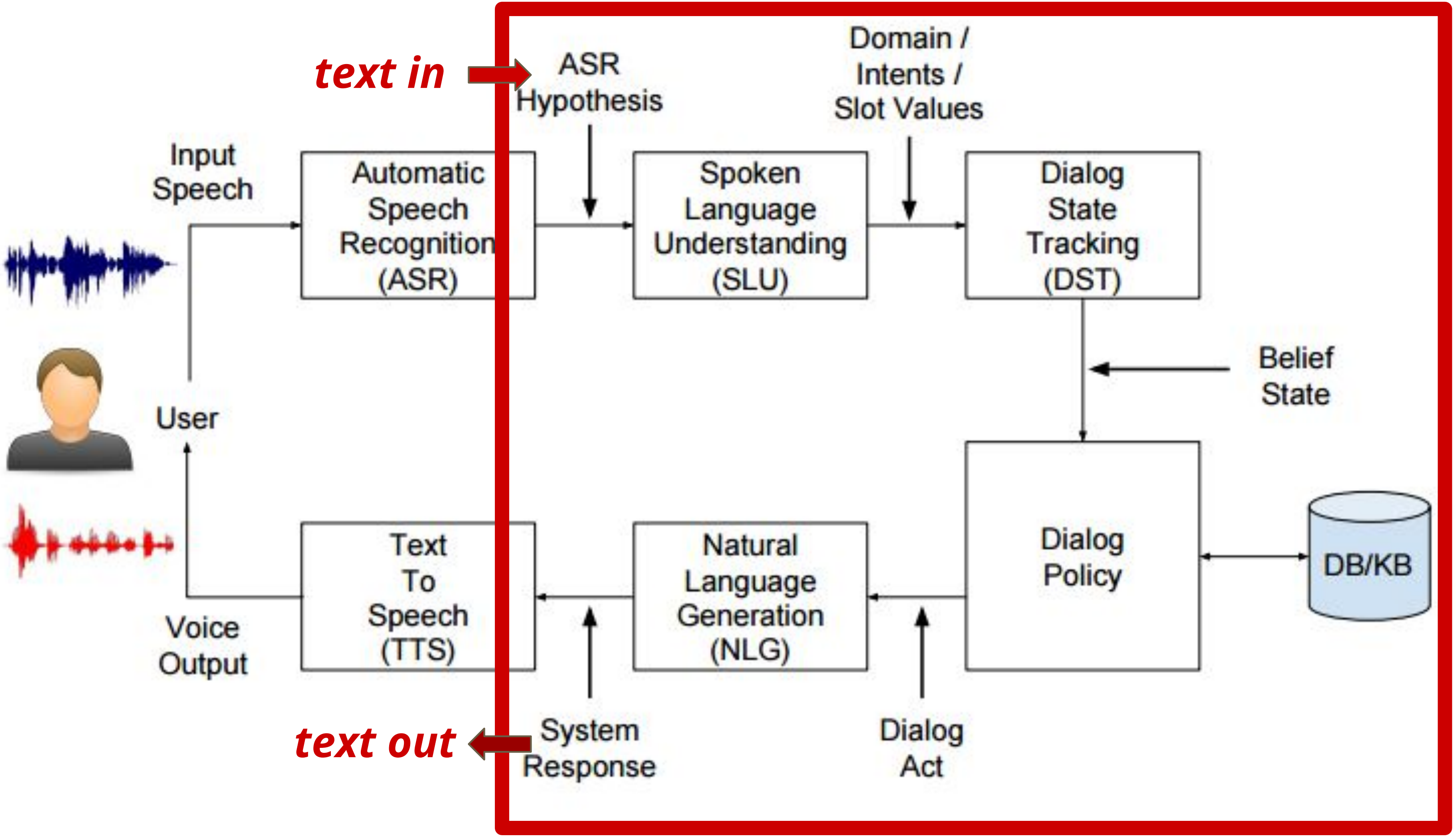
- ***"I want to chat"***
- board focus
- casual, non-goal-oriented,
  build long-term empathy, trust and companionship
- the more turns the better
- generating natural, relevant and diverse responses to make conversations engaging

RASA

Amazon Lex



Task-Oriented Dialogue System



Dialogue Example (Restaurant)

Is there a cheap place in the north part of town?

There are two restaurants that meet your specification. One is Italian and the other is Indian. Which would you prefer?

Italian, please. Can you give me the address?

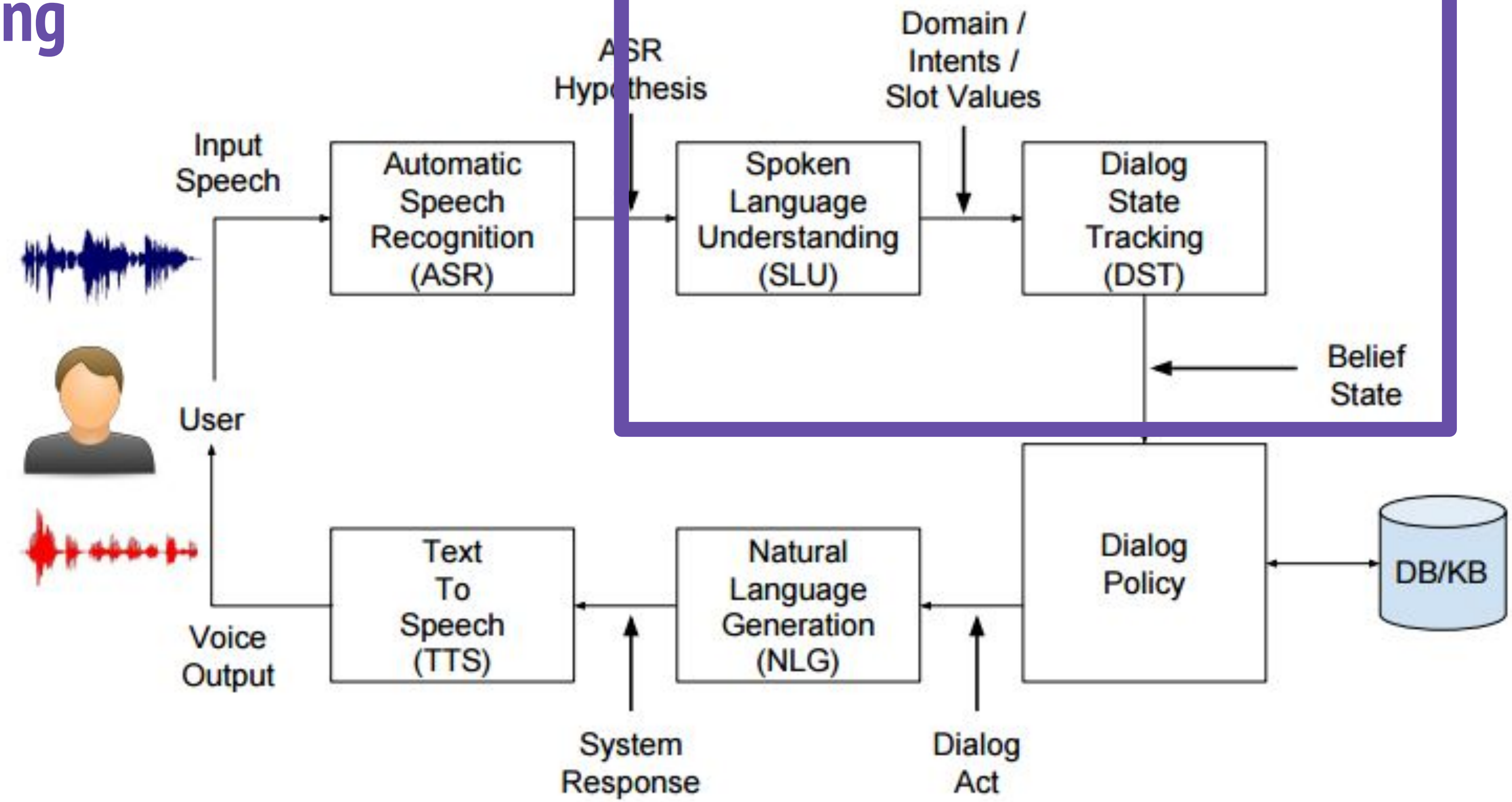
da vinci pizzeria is in the north part of town their address is 20 Milton Road Chesterton.

What is the phone number please?

da vinci pizzeria's phone number is 01223351707. Can I help you with anything else?

No that will be it. Thank you

Natural Language Understanding and Dialogue State Tracking



Dialogue State

Is there a cheap place in the north part of town?

<informational slots> {food: None} {pricerange: cheap} {area: north}

<Requestable slots>

Italian, please. Can you give me the address?

<informational slots> {food: italian} {pricerange: cheap} {area: north}

<Requestable slots> address

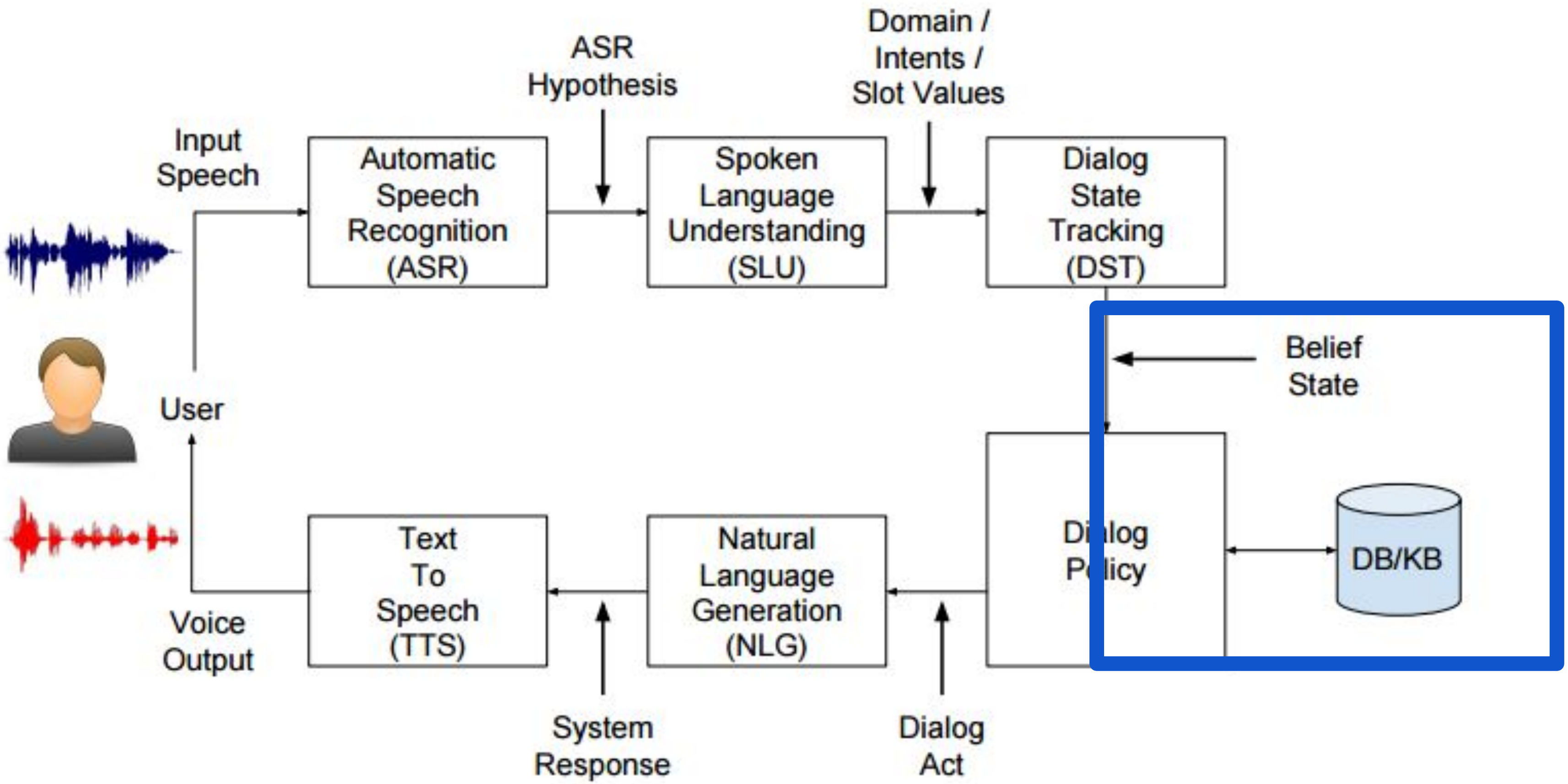
What is the phone number please?

<informational slots> {food: italian}; {pricerange: cheap}; {area: north}

<Requestable slots> phone

No that will be it. Thank you

Query Database



Database Example

id	name	food	price range	area	address	phone	postcode
19218	mahal of cambridge	indian	cheap	centre	3 - 5 Millers Yard Mill Lane	01223 360409	C.B 2, 1 R.Q
19259	da vinci pizzeria	italian	cheap	north	20 Milton Road Chesterton	01223 351707	C.B 4, 3 A.X
19257	royal spice	indian	cheap	north	Victoria Avenue Chesterton	01733 553355	C.B 4, 1 E.H
...	...	...	...	...	...	...	...

Dialogue State & DB Query Result

Is there a cheap place in the north part of town?

<informational slots> {food: None} {pricerange: cheap} {area: north}

<Requestable slots>

<DB query result> 2

Italian, please. Can you give me the address?

<informational slots> {food: italian} {pricerange: cheap} {area: north}

<Requestable slots>

<DB query result> 1

What is the phone number please?

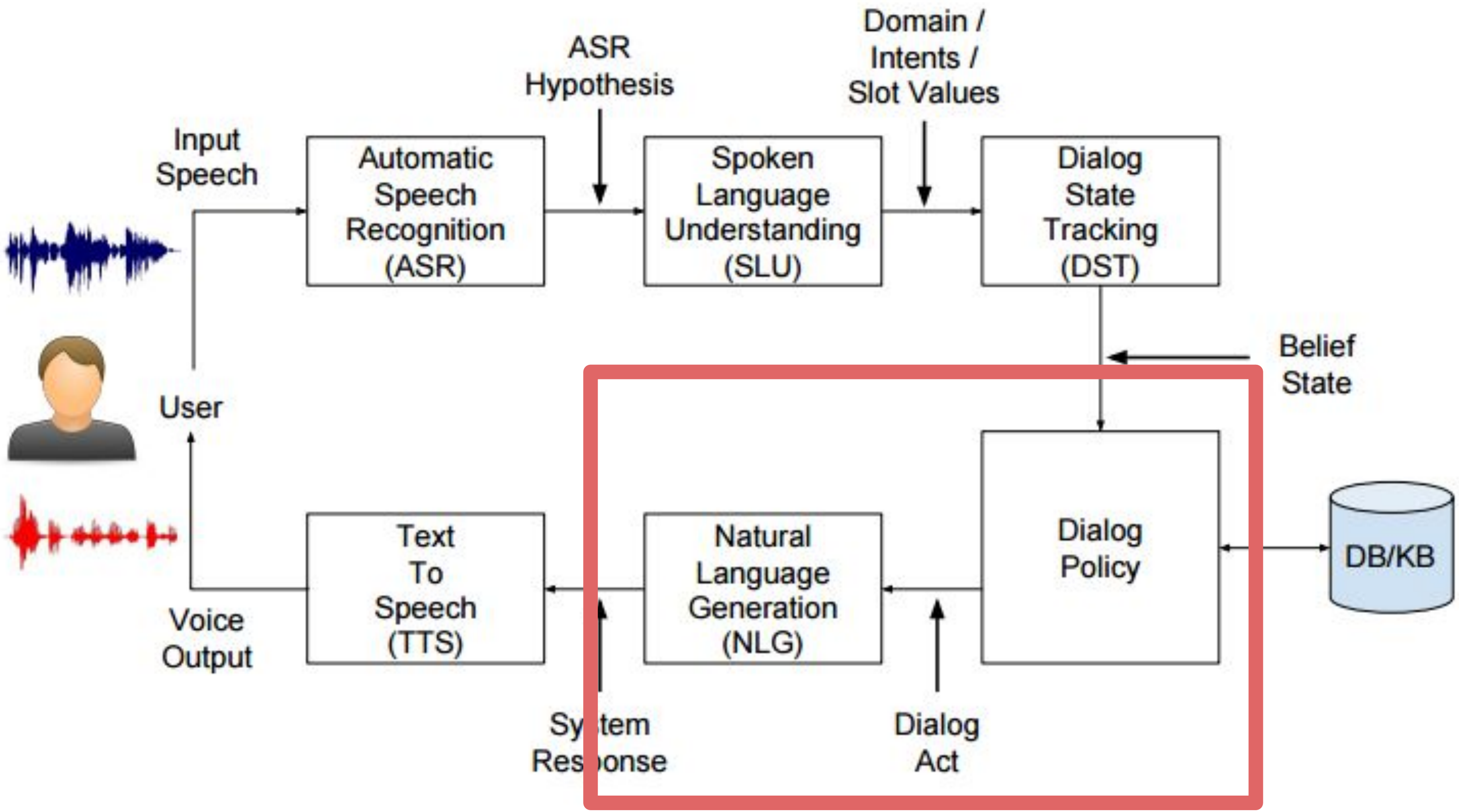
<informational slots> {food: italian}; {pricerange: cheap}; {area: north}

<Requestable slots> phone

<DB query result> 1

No that will be it. Thank you

Policy Engine and Natural Language Generation



Dialogue Act & delexicalized System Response

Is there a cheap place in the north part of town?

inform (food) multiple_choice (food)

There are two restaurants that meet your specification. One is FOOD_SLOT and the other is FOOD_SLOT. Which would you prefer?

Italian, please. Can you give me the address?

inform (address, area, name)

NAME_SLOT is in the AREA_SLOT part of town their address is ADDRESS_SLOT.

What is the phone number please?

inform (name, phone number) request (other)

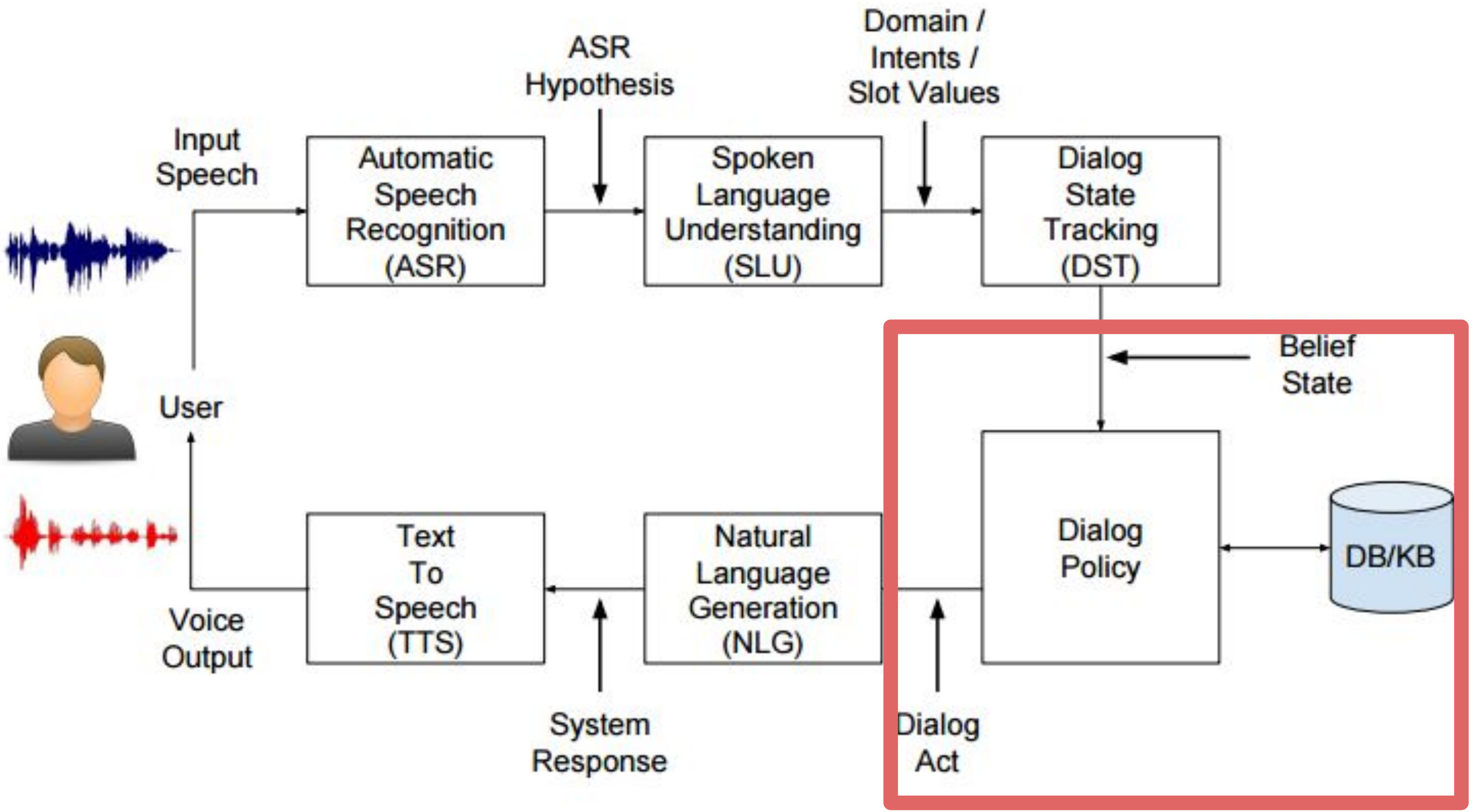
NAME_SLOT 's phone number is PHONE_SLOT. Can I help you with anything else?

No that will be it. Thank you

Roadmap

- Introduction to Task-Oriented Dialogue System
- **Modeling Multi-Action for Task-Oriented Dialogues**
  Shu et al. *EMNLP 2019*
- Flexible Structured Model for Task-Oriented Dialogues
  Shu et al. *SIGDIAL 2019, NeurIPS 2018 Conversational AI Workshop*

Policy Engine



Dialogue Act

- in terms of acts and domain specific slots.
- a key role in the quality of the interaction with the user
- influences the **efficiency** (e.g., the conciseness and smoothness) of the communication between the user and the agent

user msg	Hi! I'm looking for good thriller. Are there any playing right now?
agent msg	Yes, there are! The Witch, The Other Side of the Door, and The Boy are all thrillers. Would you like to find tickets for a showing for any of them?
agent acts	inform(moviename=The Witch, The Other Side of the Door, The Boy; genre=thriller) multiple_choice(moviename)

Single Act

disadvantage

- ***limits*** what an agent can do in a turn
- leads to ***lengthy dialogues***
- makes tracking of state and context throughout the dialogue ***harder***
- ***challenges*** users' patience

advantage

- ***easy to fine tune*** with reinforcement learning approach after supervised learning

Multi Act

advantage

- ***expands*** what an agent can do in a turn
- leads to ***efficient*** as possible
- makes tracking of state and context throughout the dialogue ***easier***

disadvantage

- ***challenge*** reinforcement learning approach

Multi-Act Prediction can be

- a **multi-label classification** problem (by ignoring sequential dependency among the acts)
- a **sequence generation**
- We propose to generate a **sequence of tuples (continue, act, slots)**
  - **maintain the dependency among the acts**
  - **reduce the recurrent steps**
  - **introduce the structure of the dialogue act into architecture**

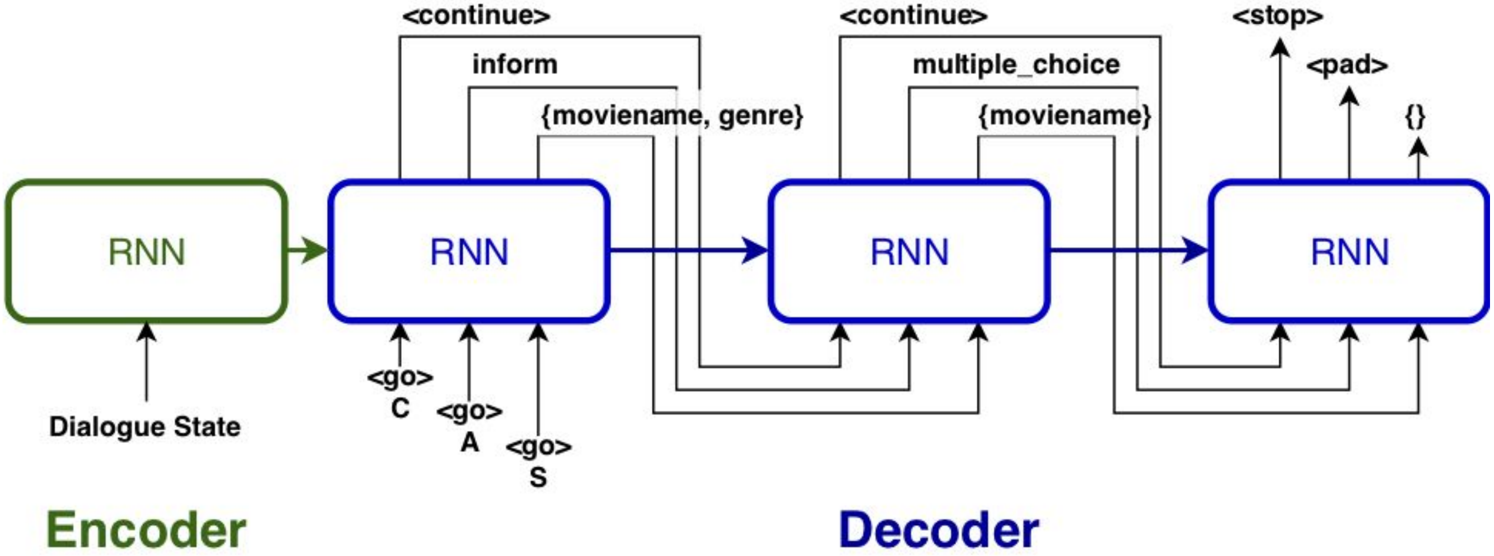
annotation	inform(moviename=The Witch, The Other Side of the Door, The Boy; genre=thriller) multiple_choice(moviename)
classification	inform+moviename, inform+genre, multiple_choice+moviename
sequence	'inform' '(' 'moviename' '=' ';' 'genre' '=' ')' 'multiple_choice' '(' 'moviename' ')' '⟨eos⟩'
cas sequence	(⟨continue⟩, inform, {moviename, genre}) (⟨continue⟩, multiple_choice, {moviename}) (⟨stop⟩, ⟨pad⟩, {})

Encoder to CAS Decoder

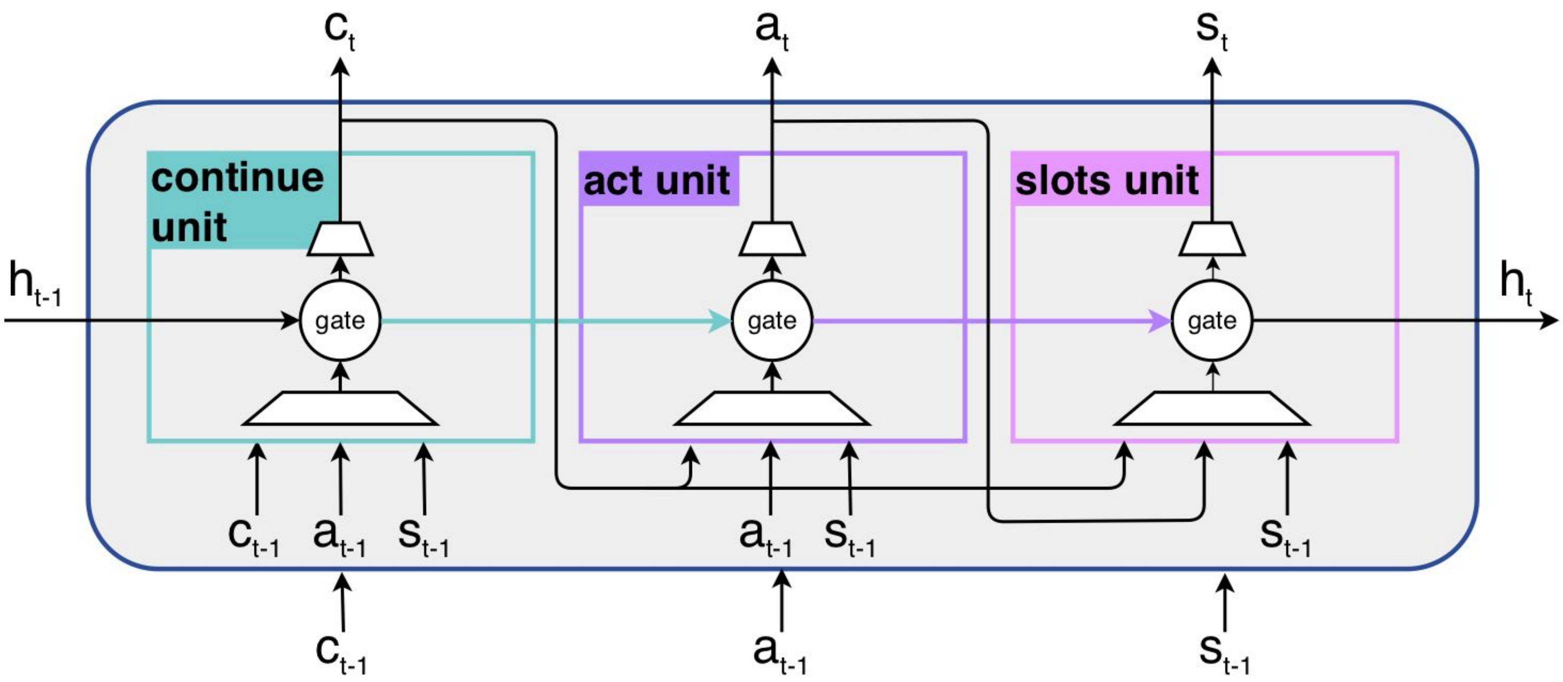
***Input*: dialogue state** (policy actions from the previous turn, user dialogue acts from the current turn, user requested slots, the user informed slots, the agent requested slots and agent proposed slots)

database queried result, we call it KB (knowledge base) vector in the paper

***Output*: a sequence of tuples (continue, act, slots)**



gated Continue Act Slot recurrent cell



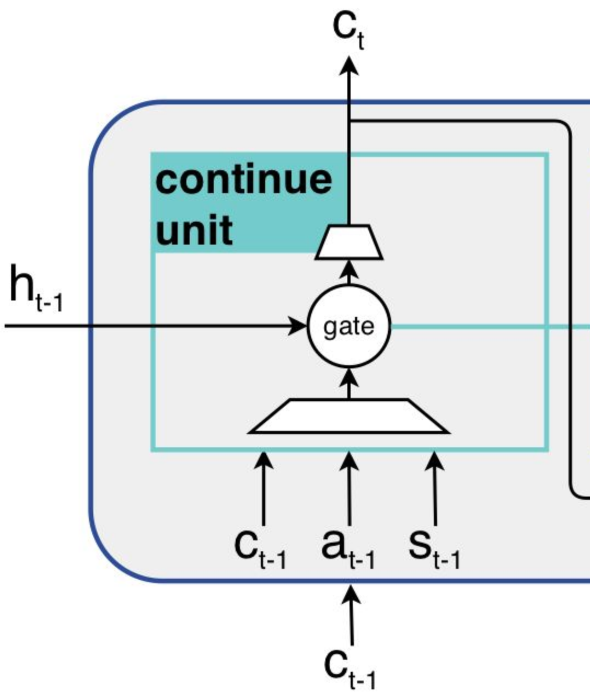
The whole gCAS decoder is recurrent-of-recurrent!

continue unit

Input: previous tuple, the KB vector, hidden state from the previous step

Output: one class from **{<continue>, <stop>, <pad>}**

$$x_t^c = W_x^c[c_{t-1}, a_{t-1}, s_{t-1}, k] + b_x^c,$$
$$g_t^c, h_t^c = \mathrm{GRU}^c(x_t^c, h_{t-1}),$$
$$P(c_t) = \mathrm{softmax}(W_g^c g_t^c + b_g^c),$$
$$\mathcal{L}^c = -\sum_t \log P(c_t).$$

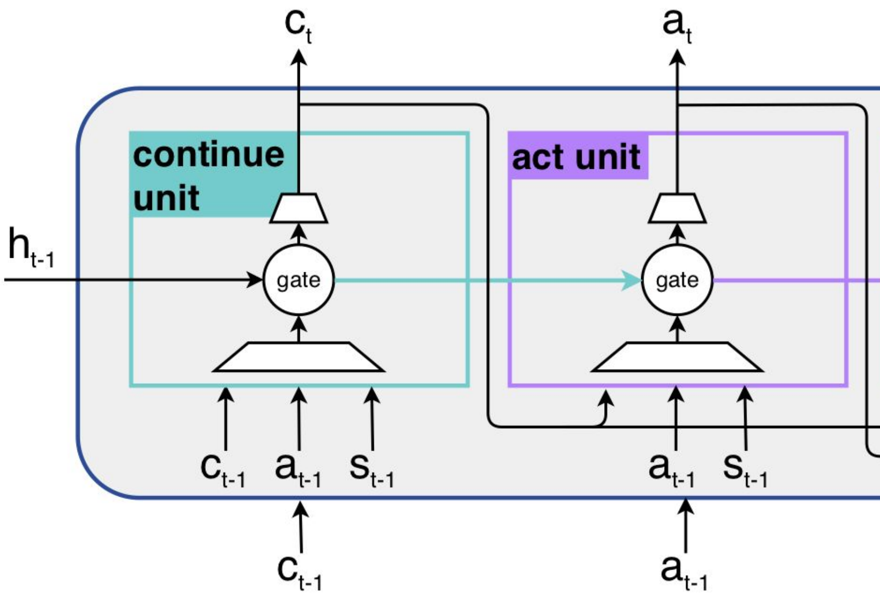


act unit

Input: previous act and slots, current continue unit's output and hidden state, the KB vector

Output: one act from act space

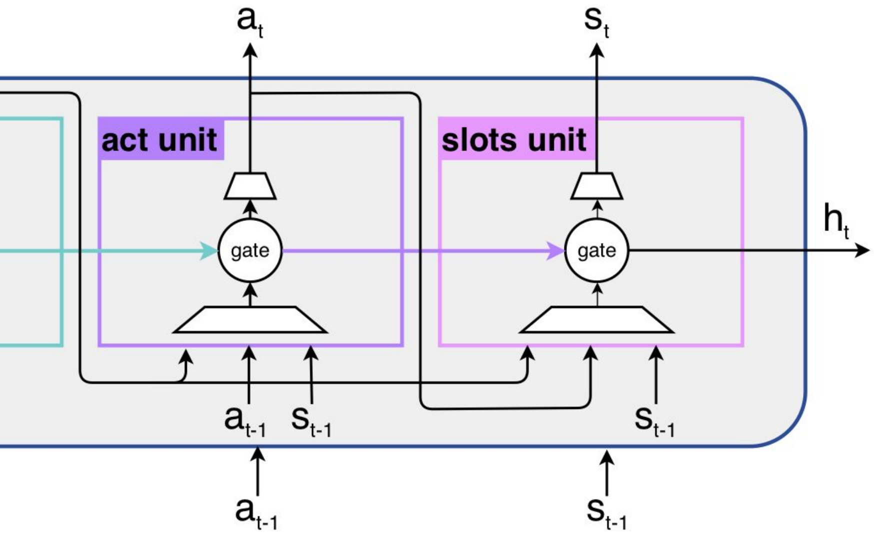
$$
\begin{aligned}
x_t^a &= W_x^a[c_t, a_{t-1}, s_{t-1}, k] + b_x^a, \\
g_t^a, h_t^a &= \mathrm{GRU}^a(x_t^a, h_t^c), \\
P(a_t) &= \mathrm{softmax}(W_g^a g_t^a + b_g^a), \\
\mathcal{L}^a &= -\sum_t \log P(a_t).
\end{aligned}
$$



slot unit

Input: previous slots, current continue unit's and act unit's outputs, the KB vector, hidden state from the act unit

Output: for each domain specific slot, it is a binary classification.



$$x_t^s = W_x^s[c_t, a_t, s_{t-1}, k] + b_x^s,$$
$$g_t^s, h_t^s = \mathrm{GRU}^s(x_t^s, h_t^a),$$
$$s_t = \mathrm{sigmoid}(W_g^s g_t^s + b_g^s),$$
$$\mathcal{L}^s = -\sum_t \sum_{i=0}^{|S|} z_t^i \log s_t^i + (1 - z_t^i) \log(1 - s_t^i).$$

Overall Loss $\mathcal{L} = \mathcal{L}^c + \mathcal{L}^a + \mathcal{L}^s$

Dataset

Microsoft Research End-to-End Dialogue Challenge

domain	total	train	valid	test	acts	slots	pairs
movie	2888	1445	433	1010	11	29	90
taxi	3093	1548	463	1082	11	23	63
restaurant	4101	2051	615	1435	11	31	91

domain & speaker	1 act	2 acts	3 acts	4 acts
movie user	9130	1275	106	11
movie agent	5078	4982	427	33
taxi user	10544	762	50	8
taxi agent	7855	3301	200	8
restaurant user	12726	1672	100	3
restaurant agent	10333	3755	403	10

Evaluation Metrics

precision, recall, F1 score of *turn-level* acts and frame

For *task completion* evaluation, **Entity F1 score** and **Success F1 score** (Lei et al., 2018) are reported.

The **Entity F1 score** (differently from the entity match rate in state tracking) compares the slots requested by the agent with the slots the user informed about and that were used to perform the KB query. We use it to measure agent performance in requesting information.

The **Success F1 score** compares the slots provided by the agent with the slots requested by the user. We use it to measure the agent performance in providing information

Baselines

- **Classification** replicates the MSR challenge (Li et al., 2018) policy network architecture: two fully connected layers. We replace the last activation from softmax to sigmoid in order to predict probabilities for each act-slot pair.

- **Seq2Seq** (Sutskever et al., 2014) encodes the dialogue state as a sequence, and decodes agent acts as a sequence with attention (Bahdanau et al., 2015).

- **Copy Seq2Seq** (Gu et al., 2016) adds a copy mechanism to Seq2Seq, which allows copying words from the encoder input.

Baselines

- **CAS** adopts a single GRU (Cho et al., 2014) for decoding and uses three different fully connected layers for mapping the output of the GRU to continue, act and slots. For each step in the sequence of CAS tuples, given the output of the GRU, continue, act and slot predictions are obtained by separate heads, each with one fully connected layer. The hidden state of the GRU and the predictions at the previous step are passed to the cell at the next step connecting them sequentially.

- **gCAS** uses our proposed recurrent cell which contains separate continue, act and slots unit that are sequentially connected.

Hyperparameter Setting

- The classification architecture has two fully connected layers of size **128**.
- The remaining models have a **hidden size of 64** and a **teacher-forcing rate of 0.5**. Seq2Seq and Copy Seq2Seq use a beam search with **beam size 10** during inference.
- **CAS and gCAS do not adopt a beam search** since their inference steps are much less than Seq2Seq methods.
- All models use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001.

Task Completion (dialogue level)

	Entity F_1			Success F_1		
	movie	taxi	restaurant	movie	taxi	restaurant
Classification	34.02	49.71	28.23	70.41	**84.45**	39.97
Seq2Seq	39.95	63.12	60.21	77.82	75.09	55.70
Copy Seq2Seq	28.04	62.95	59.14	77.59	74.58	58.74
CAS	48.02	59.16	54.70	76.81	78.89	65.18
gCAS	**50.86**	**64.00**	**60.35**	**77.95**	81.17	**71.52**

Precision, Recall, F1 score of turn-level act

	Act								
	movie			taxi			restaurant		
method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
classification	84.19	50.24	62.93	92.20	55.48	69.27	79.71	33.94	47.60
Seq2Seq	73.44	73.62	73.53	77.52	69.29	73.17	65.66	66.01	65.83
Copy Seq2Seq	67.56	73.61	70.46	73.99	69.21	71.52	64.93	65.69	65.31
CAS	70.46	76.08	73.16	79.85	72.54	76.02	65.40	72.43	68.73
gCAS	73.08	75.78	**74.41**	79.47	75.39	**77.37**	68.30	74.39	**71.22**

Precision, Recall, F1 score of turn-level act-slot pair

	Frame								
	movie			taxi			restaurant		
method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
classification	63.91	18.39	28.56	65.87	44.31	52.98	49.63	12.32	19.74
Seq2Seq	42.88	24.81	31.43	57.12	50.32	53.51	39.97	25.40	31.06
Copy Seq2Seq	41.90	23.12	29.80	51.66	50.23	50.93	36.96	27.22	31.35
CAS	43.12	31.60	36.47	51.66	54.29	52.94	33.72	25.45	29.01
gCAS	42.24	35.50	**38.58**	53.77	56.24	**54.98**	36.86	32.41	**34.49**

Generated Examples

	example 1	example 2
groundtruth	request(date; starttime)	inform(restaurantname=; starttime =) multiple_choice(restaurantname)
classification	request+date	[]
Seq2Seq	‘request’ ‘(’ ‘date’ ‘;’ ‘starttime’ ‘)’	‘inform’ ‘(’ ‘restaurantname’ ‘=’ ‘)’ ‘multiple_choice’ ‘=’ ‘restaurantname’ ‘)’
Copy Seq2Seq	‘request’ ‘(’ ‘date’ ‘=’ ‘)’	‘inform’ ‘(’ ‘restaurantname’ ‘=’ ‘;’ ‘;’, ‘;’, ‘=’, ‘;’ ‘starttime’ ‘=’ ‘)’
CAS	request {}	inform {restaurantname}
gCAS	request {date; starttime}	inform {restaurantname} multiple_choice{restaurantname}

Conclusion

We introduced a **multi-act dialogue policy model** motivated by the need for a richer interaction between users and conversation agents.

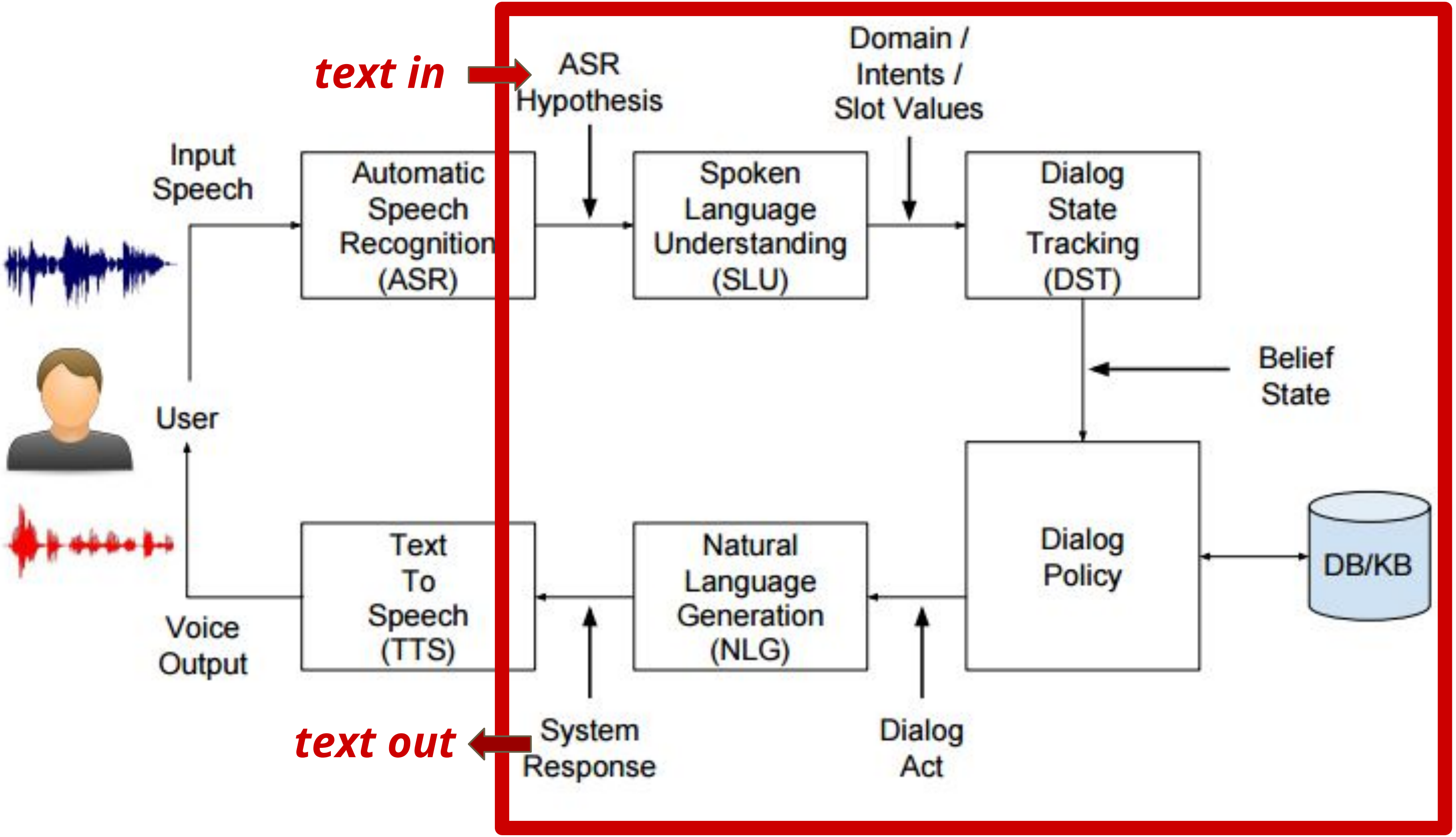
We studied classification and sequence generation methods for this task, and proposed a **novel recurrent cell, gated CAS**, which allows the decoder to output a tuple at each step.

Experimental results showed that **gCAS is the best performing model** for multi-act prediction. The CAS decoder and the gCAS cell can also be used in a user simulator and gCAS can be applied in the encoder.

Roadmap

- ➢ Introduction to Task-Oriented Dialogue System
- ➢ Modeling Multi-Action for Task-Oriented Dialogues
  Shu et al. *EMNLP 2019*
- ➢ **Flexible Structured Model for Task-Oriented Dialogues**
  Shu et al. *SIGDIAL 2019, NeurIPS 2018 Conversational AI Workshop*

Task-Oriented Dialogue System



Naive End-to-End Dialogue System

e.g. **seq2seq**

Advantages:

- simplicity

Disadvantages:

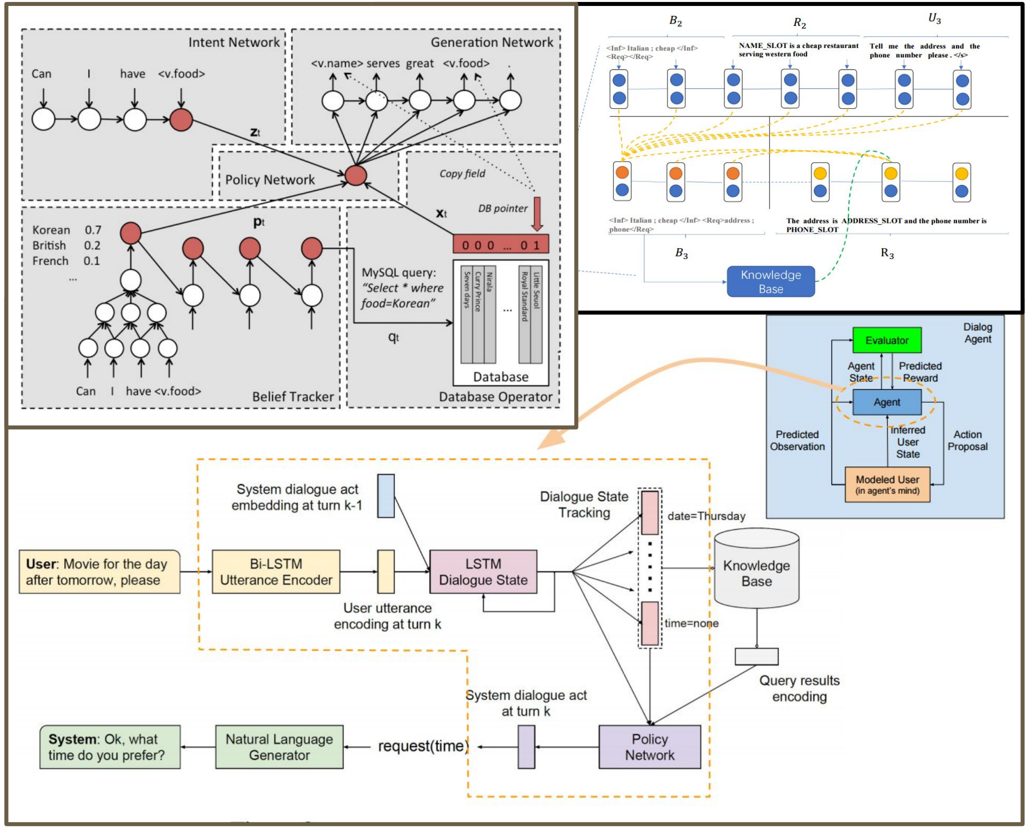
- no belief state
- no database representation
- lexicalized vocabulary (e.g. all names for all restaurant are in your vocabulary)
- no modularity (increased sample size)

Modularized End-to-End Dialogue Systems

Modules:
Natural language understanding, dialogue state tracking, knowledge base (KB) query, dialogue policy engine, response generation.

End-to-End: modules are connected and trained together with text as input and text as output.

Advantage: reduce error propagation



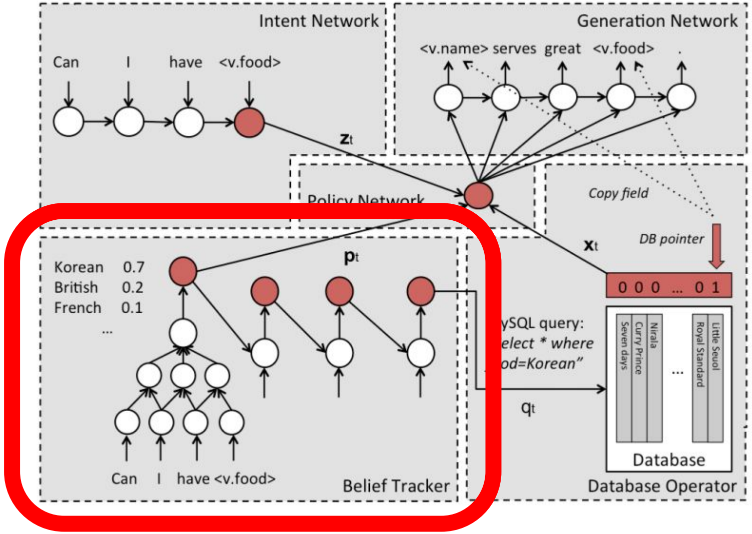
Wen et al 2017, Liu and Lane 2017, Lei et al 2018

Dialog State Tracking Module

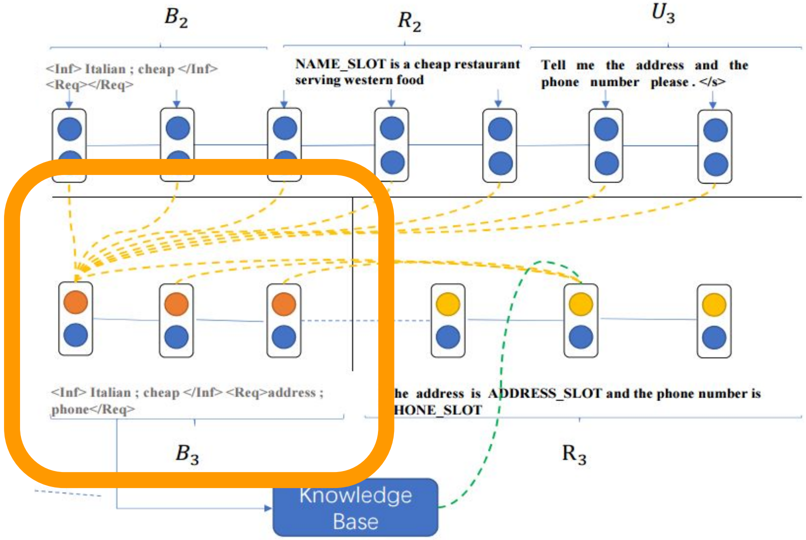
Understands user's latest intention, tracks dialog history, and updates dialog state at each turn.

The updated dialog state is used for querying the Knowledge Base (KB) and for policy engine / response generation.

Two popular approaches: ***fully-structured*** and ***free-form***.



Wen et al 2017



Lei et al 2018

Fully-Structured Approach

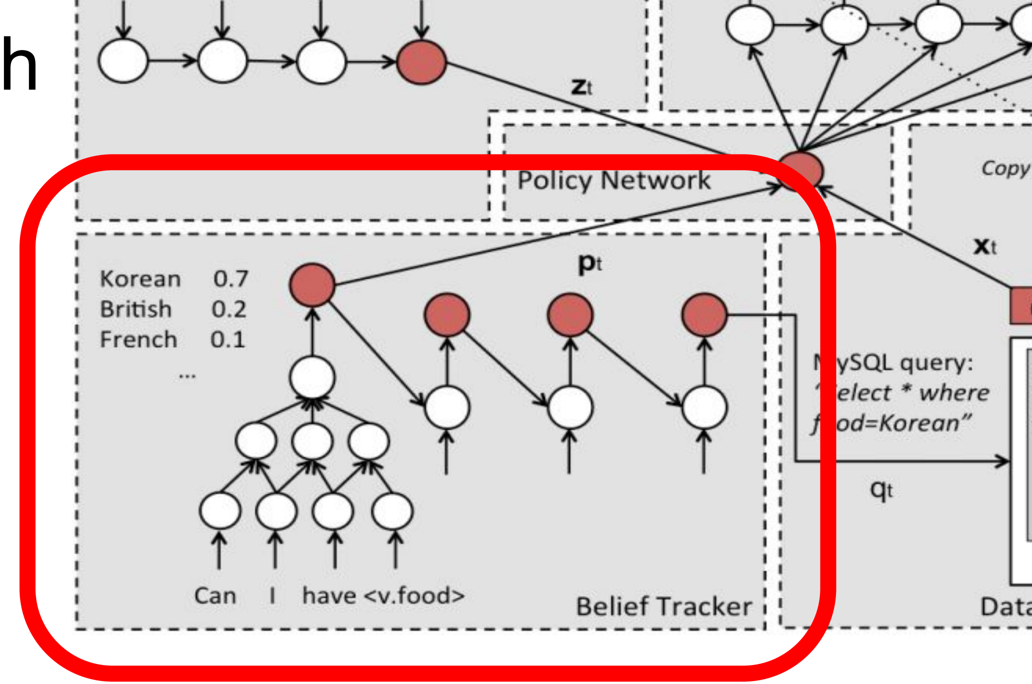
Use the **full structure of the KB**, both its schema and the values.

Assumption: the sets of informable slot values and requestable slots are fixed.

Network: multi-class classification.

Pros: value and slot are well aligned.

Cons: CANNOT adapt to dynamic KB and detect out-of-vocabulary values in the user's utterance.



Wen et al 2017

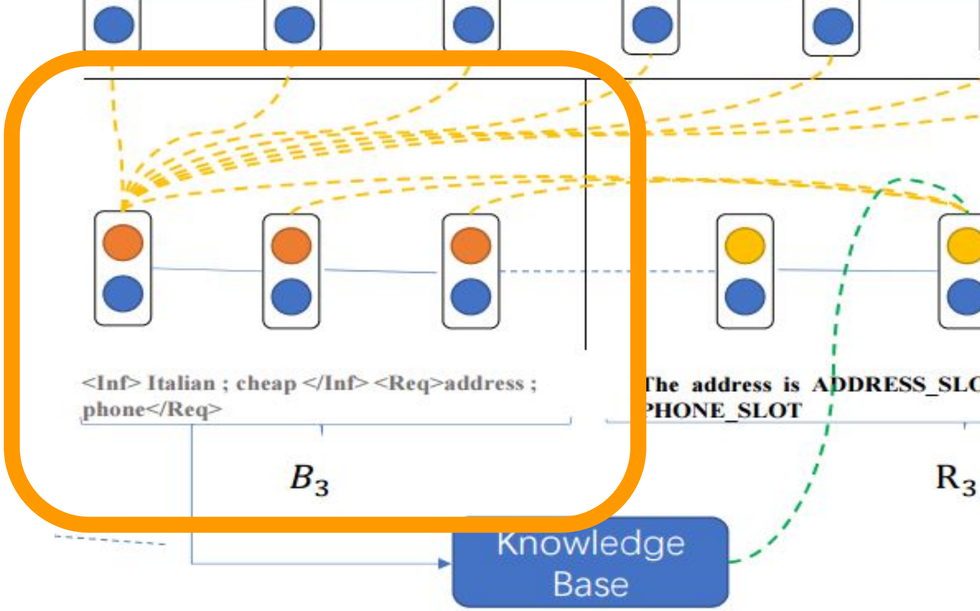
Free-Form Approach

DOES NOT exploit any information about the KB in the model architecture.

Network: sequence-to-sequence.

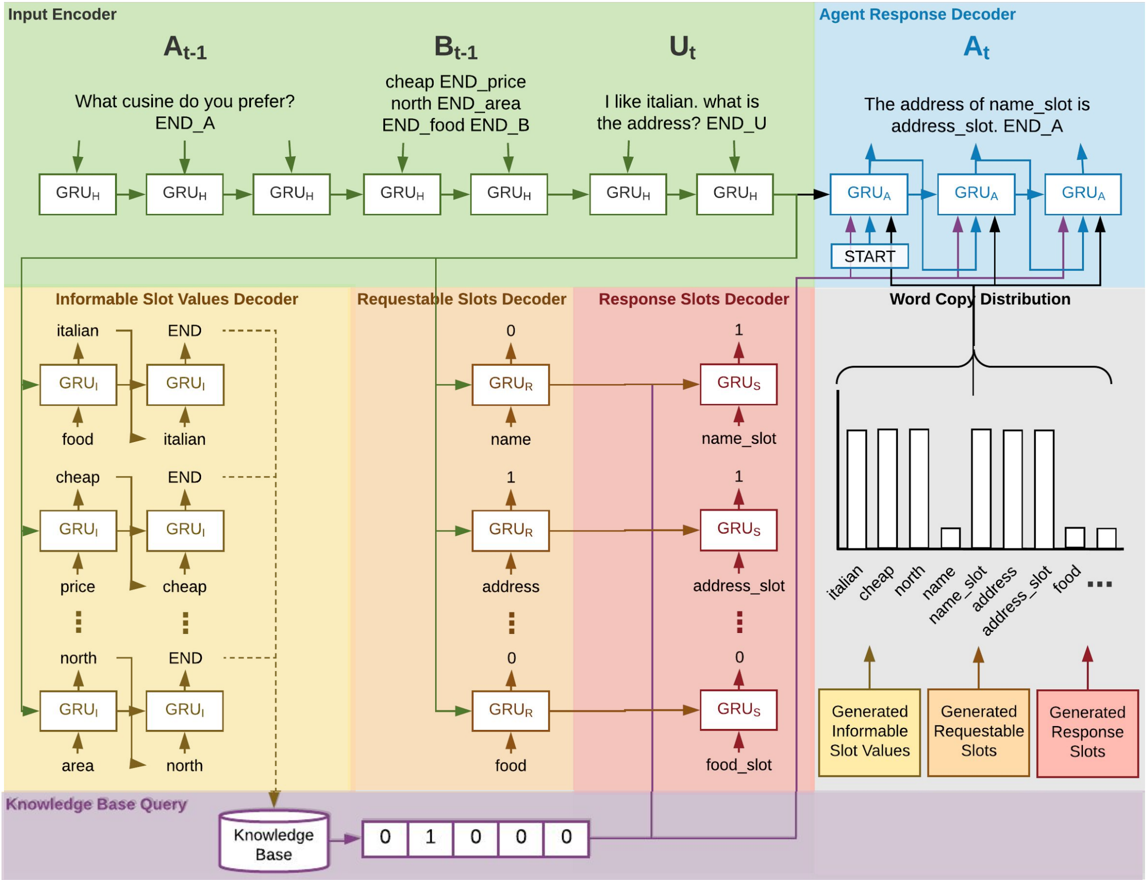
Pros:
- adaptable to new domains and changes in the content of the KB
- solves the out-of-vocabulary problem

Cons:
- value and slot are not aligned. E.g. in the travel booking system, “Chicago; Seattle”, can you tell which is the departure and which is the arrival?
- unwanted order of slots, e.g. “address; party”, “address; time; party”
- Invalid states can be generated, like including non-requestable-slot words



Lei et al 2018

Flexible-Structured Dialogue Model (FSDM)

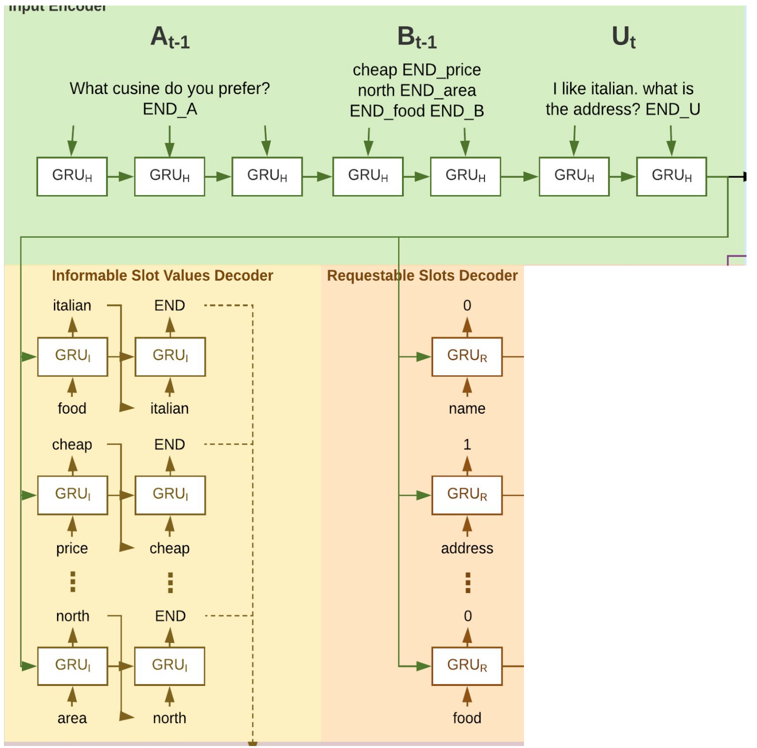


We propose: **Flexibly-Structured** DST

Use only information in the schema of the knowledge base, but not information about the values.

Architecture:

- **Informable Slot Value Decoder**: separate decoder for each informable slot (share parameters, but different start token)
- **Requestable Slot Decoder**: multi-label classifier for the requestable slots



Pros:

- slot and value are aligned
- solve the out-of-vocabulary problem
- adaptable to new domains and changes in the content of the KB
- No unwanted order of requestable slots and invalid state

Features of Flexible-Structured DST

Explicitly assign values to slots like the fully structured approach, while also preserving the capability of dealing with OOV like the free-form approach.

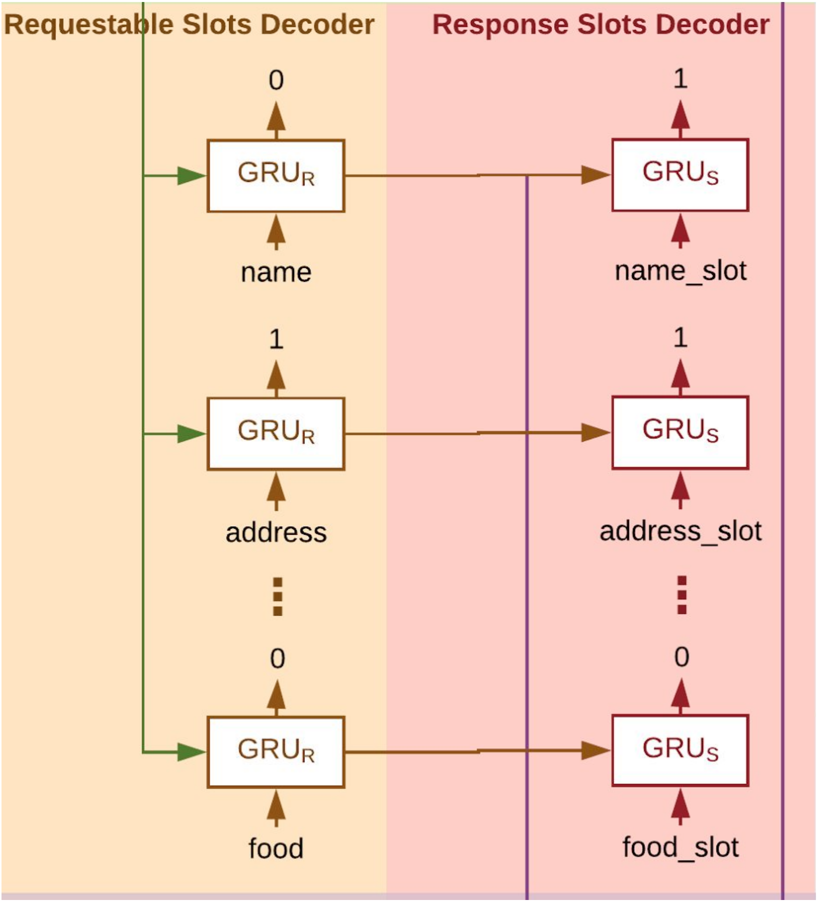
Can be applied in real-world scenarios.

It brings challenges in response generation:

(1) **Is it possible to improve the response generation quality based on Flexible-Structured DST?**

(2) **How to incorporate the output from Flexible-Structured DST for response generation?**

Solution: Response Slot Decoder



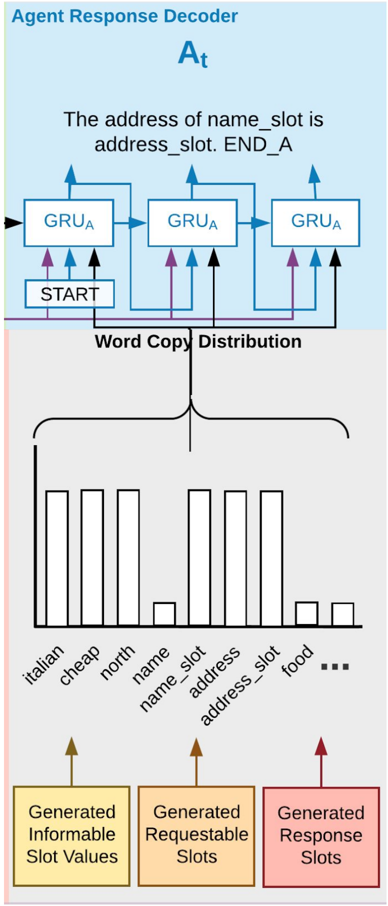
Response Slot Decoder

Response slots are the slot names that are appear in a de-lexicalized response.

Multi-label classifier is adopted for predicting which response slots will appear in the agent response.

Example:
User: request(address)
System: The address of <name_slot> is in <address_slot>

Solution: Word Copy Distribution



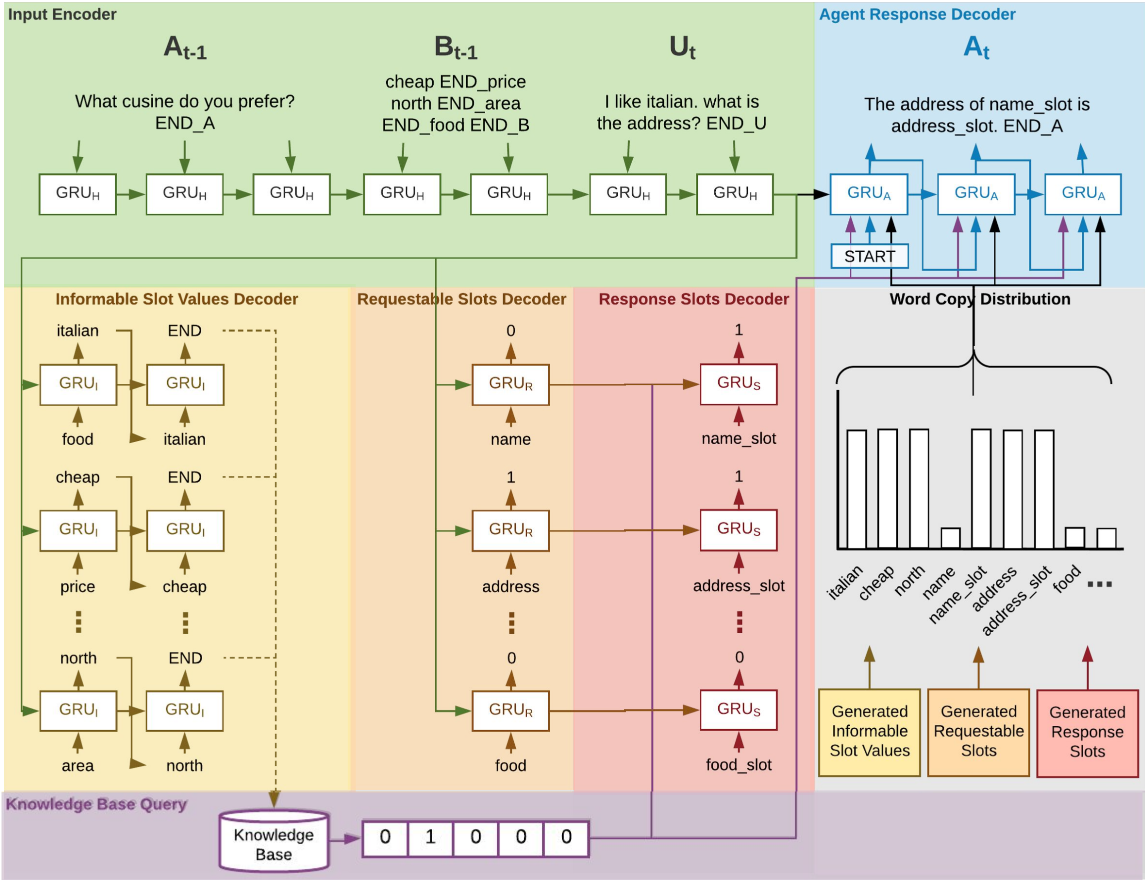
Example:
System: The address of <name_slot> is in <address_slot>

Word Copy Distribution

Increases the chance of a word in generated informable slot values, requestable slots and response slots to appear in the agent response.

Used together with copy-mechanism (Gu et al 2016).

Flexible-Structured Dialogue Model (FSDM)



Overview of FSDM

Five components that work together in an end-to-end manner

(1) The encoder encodes the agent response, the belief state, and the current user utterance

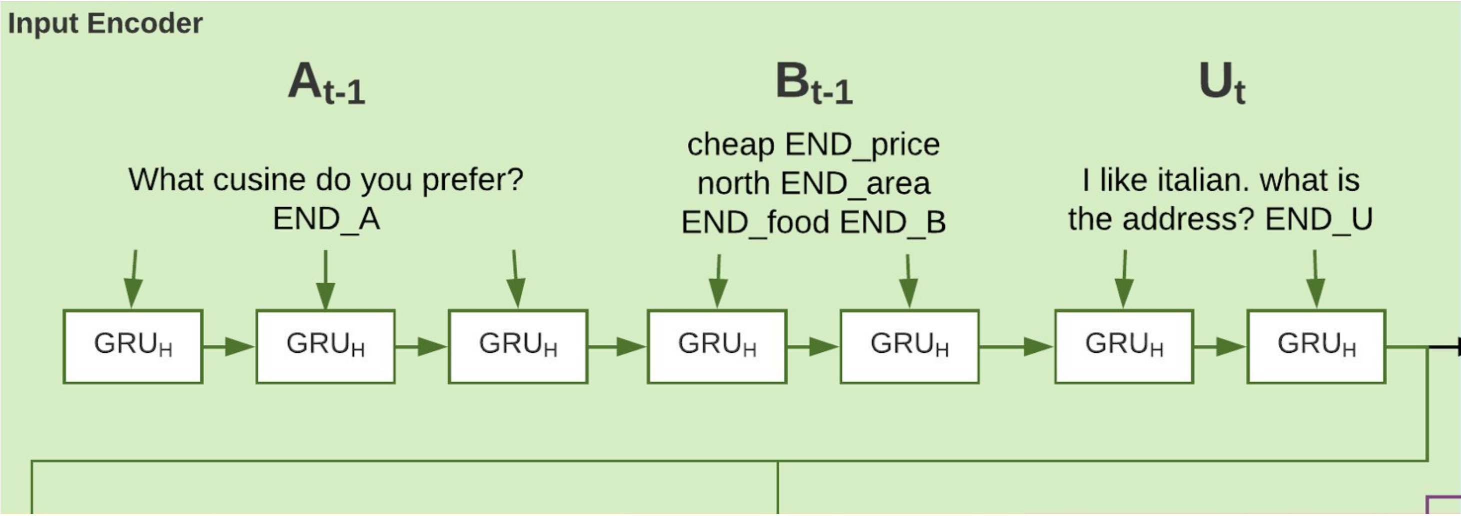
(2) The dialog state tracker contains informable slot value decoder and requestable slot binary classifier; Both take the last hidden state of encoder as the initial state

(3) Given generated informable slot values, the KB query component queries the KB and encodes the number of records returned in a one-hot vector

(4) The response slot binary classifier predicts what slots should appear in the agent response

(5) The agent response decoder takes in the KB output, a word copy probability vector, and the last hidden states of the input encoder to generate a response

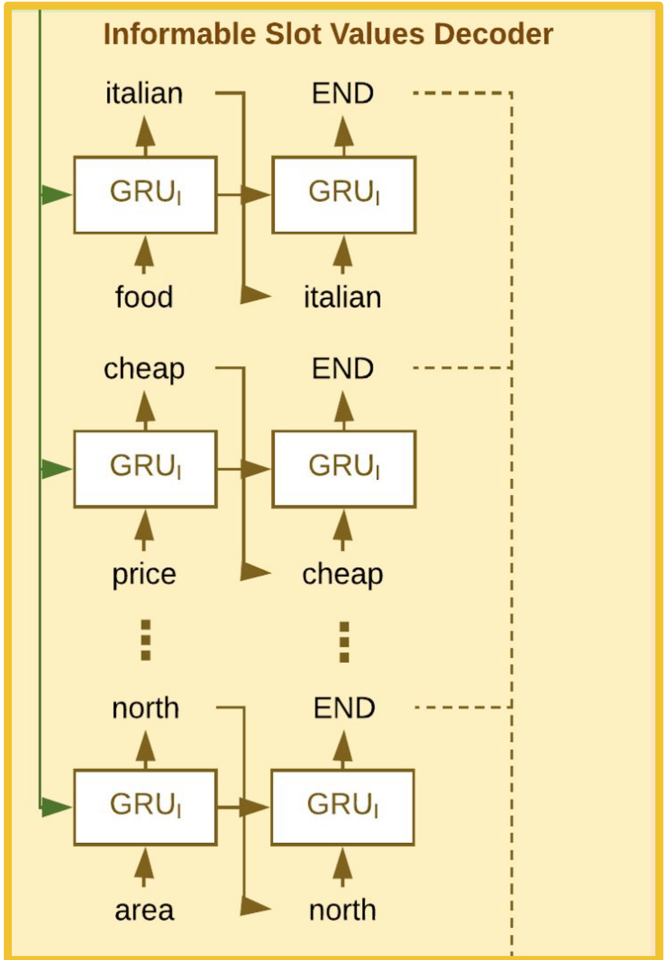
Input Encoder



Inputs: (1) the agent response A_{t-1} , (2) the dialogue state B_{t-1} from the (t-1)-th turn , (3) the current user utterance U_t .

Outputs: last hidden state of the encoder serves as the initial hidden state of the dialogue state tracker and the response decoder

Informable Slot Values Decoder

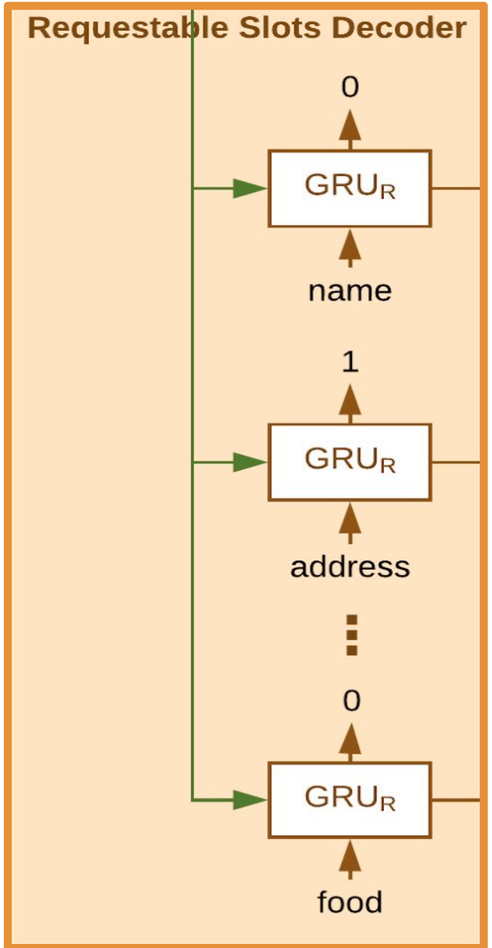


Inputs: (1) last hidden state of the encoder (2) unique start-of-sentence symbols for each slot, for example food slot's starting word is "food"

Outputs: For each slot, a sequence of words regarding this slot's value are generated. For example, the value generated for food slot is "italian END_food"

Intuition: The unique start-of-sentence symbols ensures slot and value alignment. The copy-mechanism seq2seq allows copying value directly from encoder input.

Requestable Slots Binary Classifier



Inputs: (1) last hidden state of the encoder (2) unique start-of-sentence symbols for each slot, for example food slot's starting word is "food".

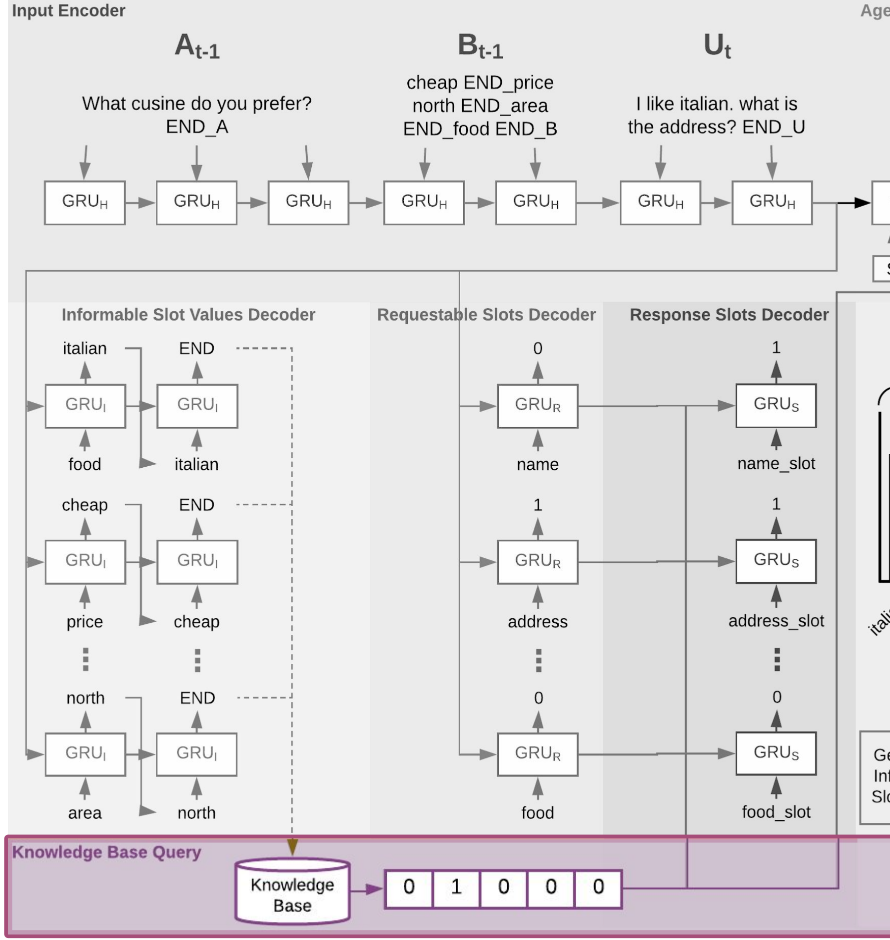
Outputs: For each slot, a binary prediction (1/0) is produced regarding whether this slot is requested by the user or not.

Note that the GRU here is only one-step. It may be replaced with any classification architecture. We choose GRU because we want to use the hidden state here as the initial state of response slot binary classifier.

Knowledge Base Query

Inputs: (1) generated informable slot values (2) Knowledge base

Outputs: a one-hot vector represents the number of records matched.

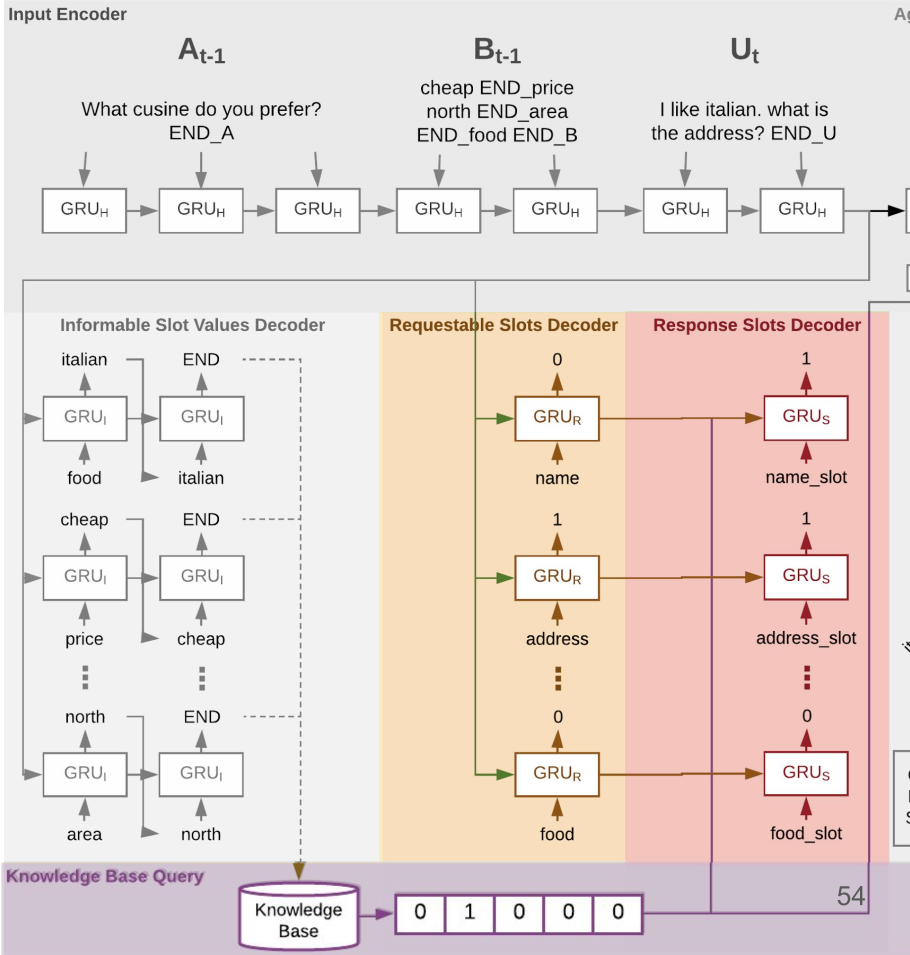


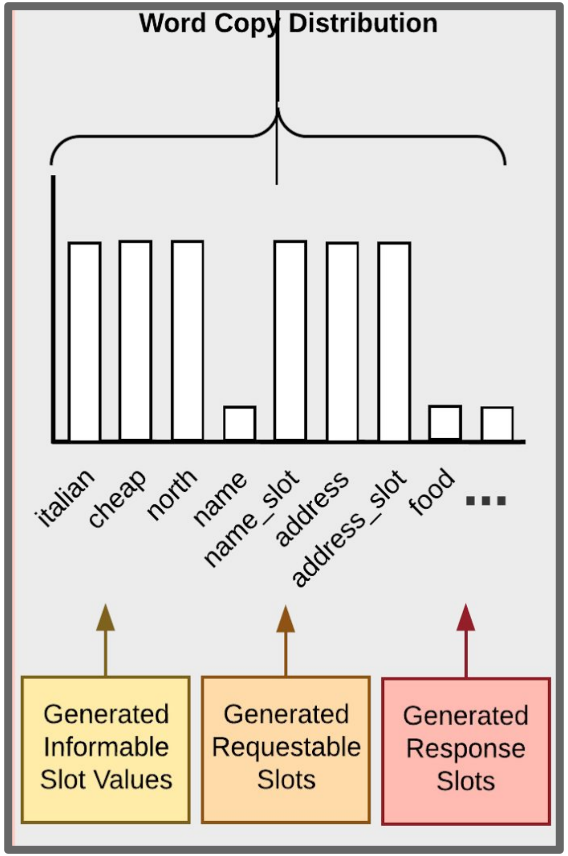
Response Slots Binary Classifier

Inputs: (1) KB query result (2) hidden states of requestable slot binary classifier

Outputs: For each response slot, a binary prediction (1/0) is produced regarding whether this response slot appears in the agent response or not.

Motivation: incorporate all the relevant information about the retrieved entities and the requested slots into the response





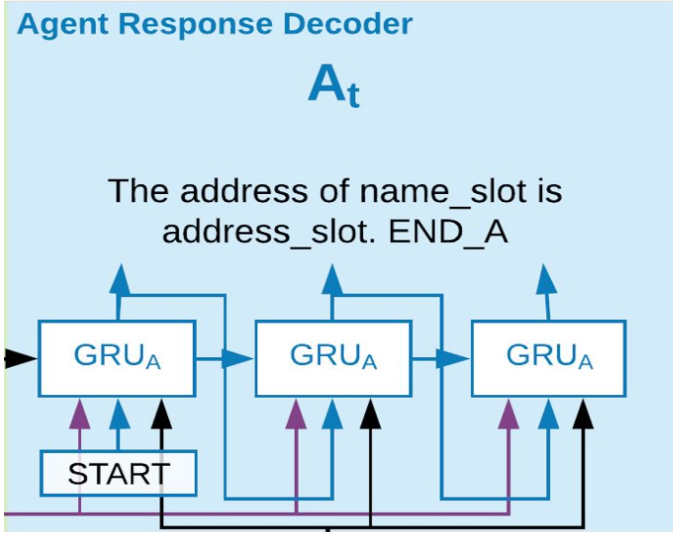
Motivation: The canonical copy mechanism only takes a sequence of word indexes as inputs, but does not accept the multiple Bernoulli distributions we obtain from binary classifiers.

Inputs: prediction from (1) informable slot value decoders, (1) requestable slot binary classifier, (3) response slot binary classifiers.

Outputs: if a word is a requestable slot or a response slot, the probability is their binary classifier output; if a word appears in the generated informable slot values, its probability is equal to 1; for all other words, 0.

$$\mathcal{P}^{\mathcal{C}}(w) = \begin{cases} y^{k^R}, & \text{if } w = k^R, \\ y^{k^S}, & \text{if } w = k^S, \\ 1, & \text{if } w \in I_t, \\ 0, & \text{otherwise}, \end{cases}$$

Agent Response Decoder



Inputs: (1) last hidden state of encoder (2) KB query results result (3) Word copy distribution

Outputs: a de-lexicalized agent response

Final Loss

$$\mathcal{L} = \underbrace{\boxed{\alpha^I \mathcal{L}^I}}_{\text{Informable slot values}} + \underbrace{\boxed{\alpha^R \mathcal{L}^R}}_{\text{Requestable slots}} + \underbrace{\boxed{\alpha^S \mathcal{L}^S}}_{\text{Response slots}} + \underbrace{\boxed{\alpha^A \mathcal{L}^A}}_{\text{Agent response}}$$

Experiment Setting

Dataset:
Cambridge Restaurant dataset (CamRest) (Wen et al 2016)
Stanford in-car assistant dataset (KVRET) (Eric et al 2017)

Evaluation Metrics:
For belief state tracking, **precision**, **recall**, and **F_1 score** of informable slot values and requestable slots.

For task completion evaluation, the **Entity Match Rate (EMR)** and **Success F_1 score ($SuccF_1$)** are reported.

BLEU is applied to the generated agent responses for evaluating language quality.

Experiment Setting-Baselines

NDM (Wen et al 2016) proposes a modular end-to-end trainable network. It applies de-lexicalization on user utterances and responses. (fully structured)

LIDM (Wen et al 2017) improves over NDM by employing a discrete latent variable to learn underlying dialogue acts. This allows the system to be refined by reinforcement learning.(fully structured)

KVRN (Eric et al 2017) adopts a copy-augmented Seq2Seq model for agent response generation and uses an attention mechanism on the KB. It does not perform belief state tracking.(no DST)

TSCP/RL (Lei et al 2018) is a two-stage CopyNet which consists of one encoder and two copy-mechanism-augmented decoders for belief state and response generation. **TSCP** includes further parameter tuning with reinforcement learning to increase the appearance of response slots in the generated response. (free-form)

Turn-Level Dialogue State Tracking Result

Dataset	CamRest					
Method	Inf P	Inf R	Inf F_1	Req P	Req R	Req F_1
TSCP/RL†	0.970	0.971	0.971	0.983	0.935	0.959
TSCP†	0.970	0.971	0.971	0.983	0.938	0.960
FSDM/Res	0.979	0.984	0.978	0.994	0.947	0.967
FSDM	**0.983***	**0.986***	**0.984***	**0.997***	**0.952**	**0.974***

Dataset	KVRET					
Method	Inf P	Inf R	Inf F_1	Req P	Req R	Req F_1
TSCP/RL†	**0.936**	0.874	0.904	0.725	0.485	0.581
TSCP†	0.934	0.890	0.912	0.701	0.435	0.526
FSDM/Res	0.918	0.930	0.925	0.812	0.993	0.893
FSDM	0.92	**0.935***	**0.927***	**0.819***	**1.000***	**0.900***

Dialogue-Level Task Completion Result

Dataset	CamRest			KVRET		
Method	BLEU	EMR	$SuccF_1$	BLEU	EMR	$SuccF_1$
NDM	0.212	0.904	0.832	0.186	0.724	0.741
LIDM	0.246	0.912	0.840	0.173	0.721	0.762
KVRN	0.134	-	-	0.184	0.459	0.540
TSCP	0.253	0.927	0.854	**0.219**	0.845	0.811
TSCP/RL †	0.237	0.915	0.826	0.195	0.809	0.814
TSCP†	0.237	0.913	0.841	0.189	0.833	0.81
FSDM/St	0.245	-	0.847	0.204	-	0.809
FSDM/Res	0.251	0.924	0.855	0.209	0.834	0.815
FSDM	**0.258***	**0.935***	**0.862***	0.215	**0.848***	**0.821***

Example of generated dialogue state and response
(calendar scheduling domain)

user msg	what is the date and time of my next meeting and who will be attending it ?
	belief state
GOLD	informable slot (event=meeting), requestable slot (date, time, party)
TSCP	'meeting' '⟨EOS_Z1⟩' 'date' ';' 'party'
FSDM	event=meeting date=True time=True party = True
	agent response
GOLD	your next meeting is with party_SLOT on the date_SLOT at time_SLOT.
TSCP	your next meeting is at time_SLOT on date_SLOT at time_SLOT .
FSDM	you have a meeting on date_SLOT at time_SLOT with party_SLOT

Conclusions

FSDM: novel end-to-end architecture with flexibly-structured model for task-oriented dialogue.

Uses the structure in the schema of the KB to make architectural choices that introduce inductive bias and address the limitations of fully structured and free-form methods.

The experiment suggests that this architecture is **competitive with SOTA models**, while being applicable in **real-world scenarios**

Reference

Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In Proceedings of the NAACL-HLT.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end train-able task-oriented dialogue systems. In NAACL.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In ACL.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In ACL (1). The Association for Computer Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkˇsi´c, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017b. A network-based end-to-end trainable task-oriented dialogue system. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 438–449. ACL.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017a. Latent intention dialogue models. In ICML, volume 70 of Proceedings of Machine Learning Research, pages 3732–3741. PMLR.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In SIGDIAL Conference, pages 37–49. Association for Computational Linguistics.