Dialogue State Tracking: past, present, and future

–

Léo Jacqmin
PhD candidate at Aix-Marseille University and Orange

Dialogue Systems

From fiction to reality

- in popular culture
  - 2001: A Space Odyssey (1968)
  - Her (2013)
- today's adoption
  - virtual assistants (Alexa, Siri et al.)
  - chatbots

Dialogue Systems

Conversational agents: long-standing goal of AI

- Turing test (1950)
- rule-based chatbots: ELIZA (1964)

Motivation

- language/speech: natural interface for human-computer interactions
- open-domain dialogue vs. task-oriented dialogue

Different Goals

Task-oriented dialogue

- designed for specific domains or tasks, e.g. hotel reservation, customer service, or technical support
- shorter is better

Open-domain dialogue

- chit-chat
- maximize long-term user engagement

Different Approaches

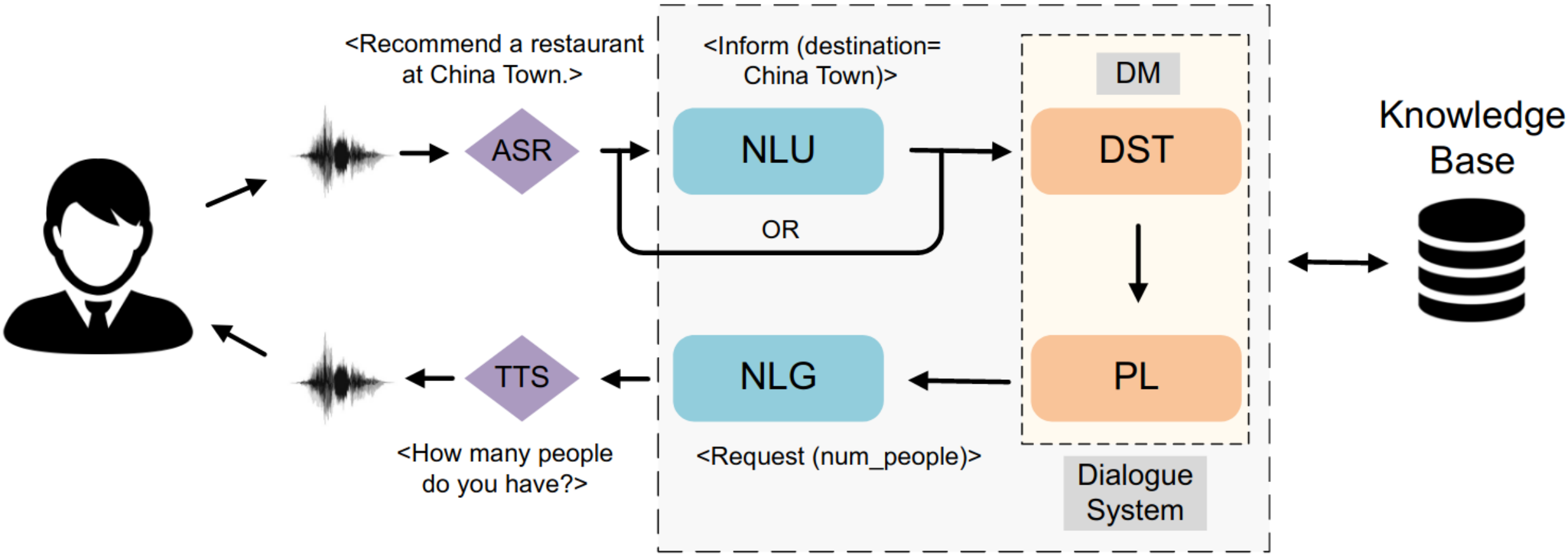
Task-oriented dialogue

- modular system: hand-crafted or trained on task-specific labeled data
- pre-defined task-specific schema: set of user intents, each intent defines a set of dialog acts, slot-value pairs

Open-domain dialogue

- end-to-end neural response generation
- grounding on open-domain knowledge

Task-oriented Dialogue Systems


<Recommend a restaurant at China Town.>
ASR
<Inform (destination= China Town)>
NLU
OR
DM
DST
PL
NLG
TTS
<How many people do you have?>
<Request (num_people)>
Dialogue System
Knowledge Base

Dialogue State

User: hello, i'm looking for a restaurant with fair prices

Dialogue State : `Inform(price range = moderate)`

Sys: There are 31 places with moderate price range. Can you please tell me what of food you would like?

User: well I want to eat in the North, what's up that way?

Dialogue State: `Inform(price range=moderate, area=north)`

Sys: I have two options that fit that description, Golden Wok chinese restaurant and The Nirala which serves Indian food. Do you have a preference?

User: Can I have the address and phone number for the Golden Wok chinese restaurant?

Dialogue State: `Inform(price range=moderate, area=north)`
`request(address, phone number)`

Dialogue State Tracking

Estimate dialogue state s_t as slot-value pairs at turn t

- Turn-level vs. dialogue level prediction

Challenging task

- large variation in the users
- noisy environment

➡ Difficult to understand what the user wants but essential for successful dialogue

DST (rough) Timeline

1. Rule-based systems (pre-2000)
2. Generative approaches (2000s)
   - $b(s_t)$
3. Discriminative approaches (2010s)
   - $P(s_t|o_1, \ldots, o_t)$
4. Neural approaches (~2015 onwards)

Rule-based Systems

Update rule maps from an existing state s_{t-1} and the 1-best SLU result o_t to a new state s_t

- requires no data to implement
- accessible method to incorporate knowledge of the dialog domain

Shortcomings

- costly and difficult to maintain
- unable to make use of the entire ASR N-best or SLU M-best lists
  ➡ do not account for uncertainty

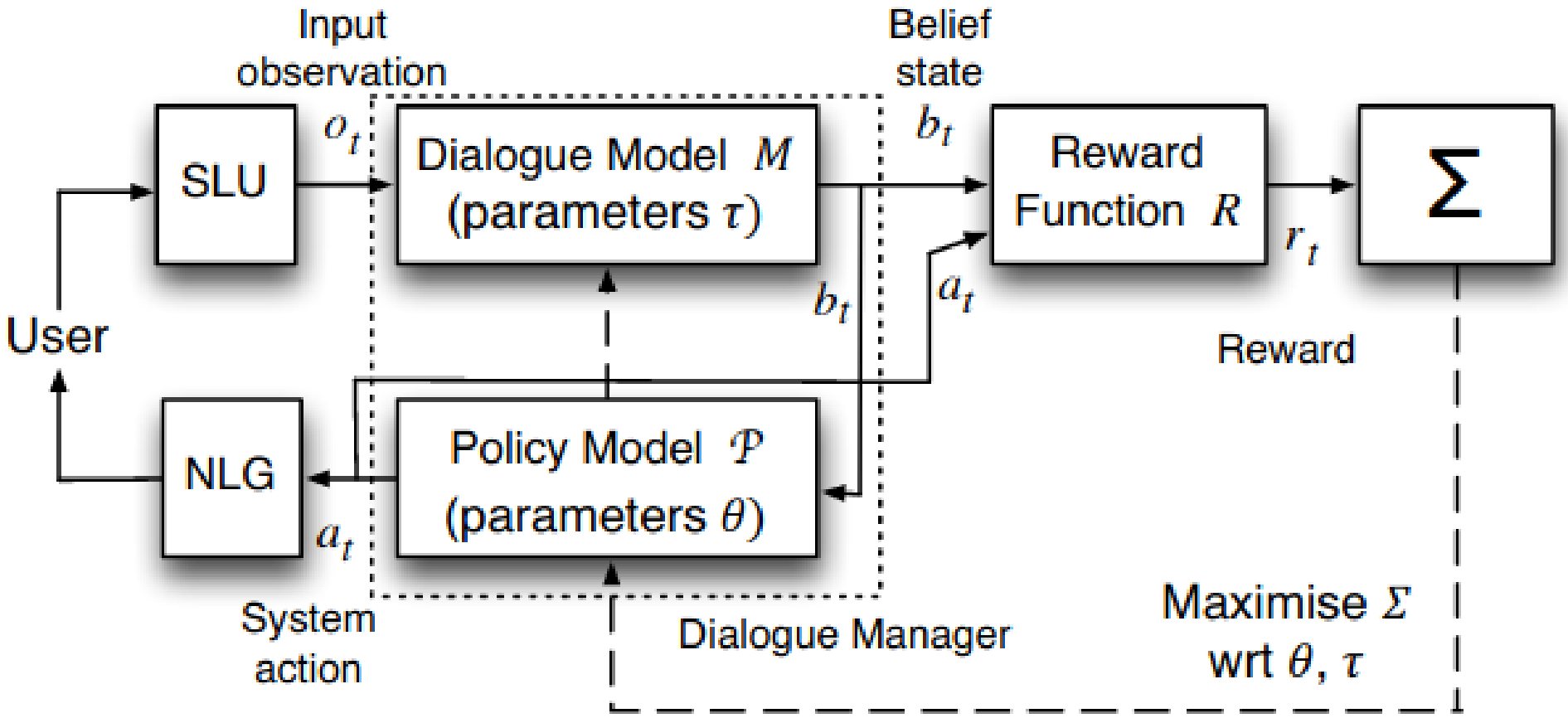
Generative Approaches

Model dialogue as a **Partially Observable Markov Decision Process**

- maintain a Markov dialogue state in each turn and choose next action based on this state
- model observations from the SLU jointly with the dialogue state using Bayesian inference

Advantages

- reduce cost of hand-crafting complex dialogue managers
- robustness against the errors created by speech recognition



POMDP-Based Statistical Spoken Dialog Systems: A Review (Young et al., 2013)

Discriminative Approaches

Directly model the class posteriors given the observations

- $b(s_t) = P(s_t | f_1, \ldots, f_t)$, where $f_1, \ldots, f_t$ are features extracted from the ASR, SLU, and dialog history

Main appeal

- any classification paradigm can be applied once the sequence of observations $f_1, \ldots, f_t$ is turned into a fixed-length summary representation

Separate SLU

- SLU decoders to detect slot-value pairs expressed in ASR output
- DST model combines this information with the past dialogue context to update the belief state

Joint SLU and DST

- word-based DST paradigm
- ASR predictions as input and belief states as output

Datasets and Evaluation Metrics

Dialogue State Tracking Challenge (DSTC)

First shared testbed and evaluation metrics for DST (2013)

- Each instance
  - released a public corpus of transcribed and labeled dialogs along with baseline trackers and evaluation tools
  - explored a new aspect of DST
- rebranded as Dialog System Technology Challenges since DSTC-6

The Dialog State Tracking Challenge Series: A Review (Williams et al., 2016)

Data Collection

Simulation-based	Wizard-of-Oz
machine-to-machine	human-to-human (crowd-sourced)
restricted task complexity and language diversity	natural and diverse dialogs
annotations obtained automatically	difficult data annotation

Metric	Datasets							
	DSTC2	**DSTC3**	**WoZ2.0**	**MultiWoZ**	**Frames**	**SGD**	**M2M**	**TreeDST**
# Dialogues	3235	2236	1200	10438	1369	22825	120000	27280
# Turns	51002	35723	8824	143048	19986	463282	1661536	167507*
Avg. turns / dial.	15.77	15.98	7.35	13.7	14.60	20.30	13.85	6.14*
Avg. tokens / turn	8.47	10.82	11.27	13.18	12.60	9.86	9.96	7.59*
# Unique tokens	1178	1873	3562	30245	13864	45578	2315	7936*
# Slots	8	13	7	29	60	339	5	289
# Values	85	118	88	2180	4508	25123	92	5687

Evaluation Metrics

- Joint Goal Accuracy
- Requested Slots F1
- Time Complexity

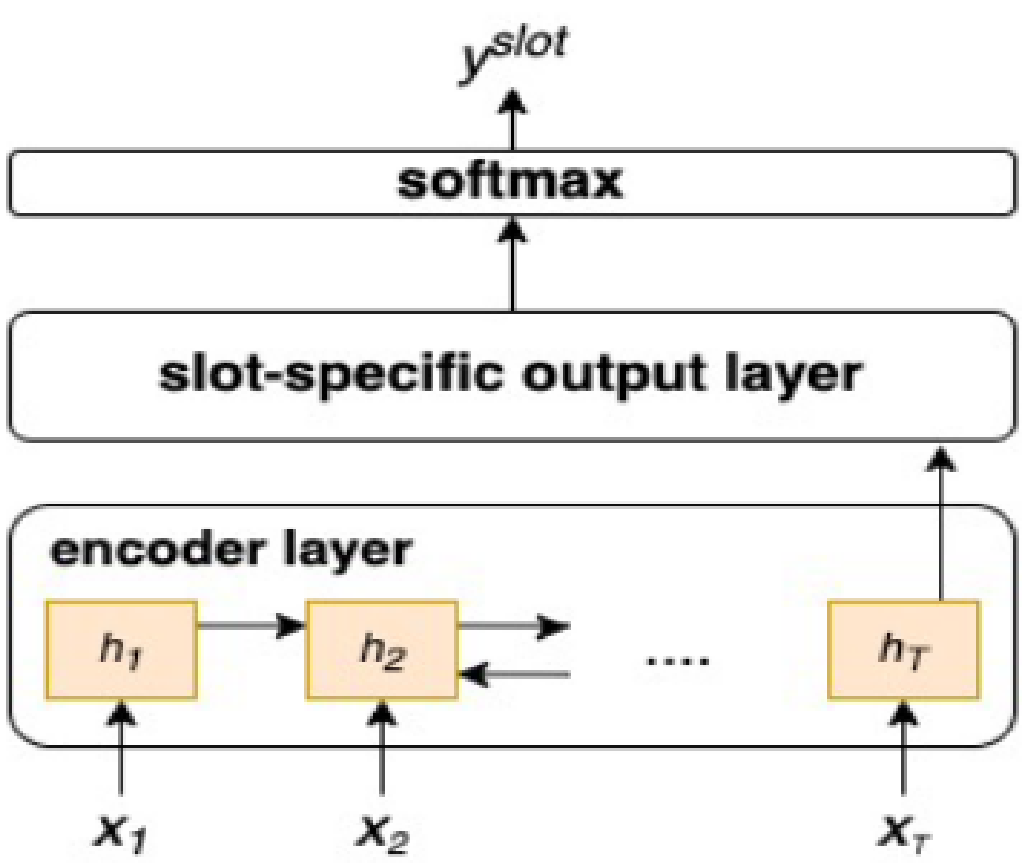
Neural Approaches

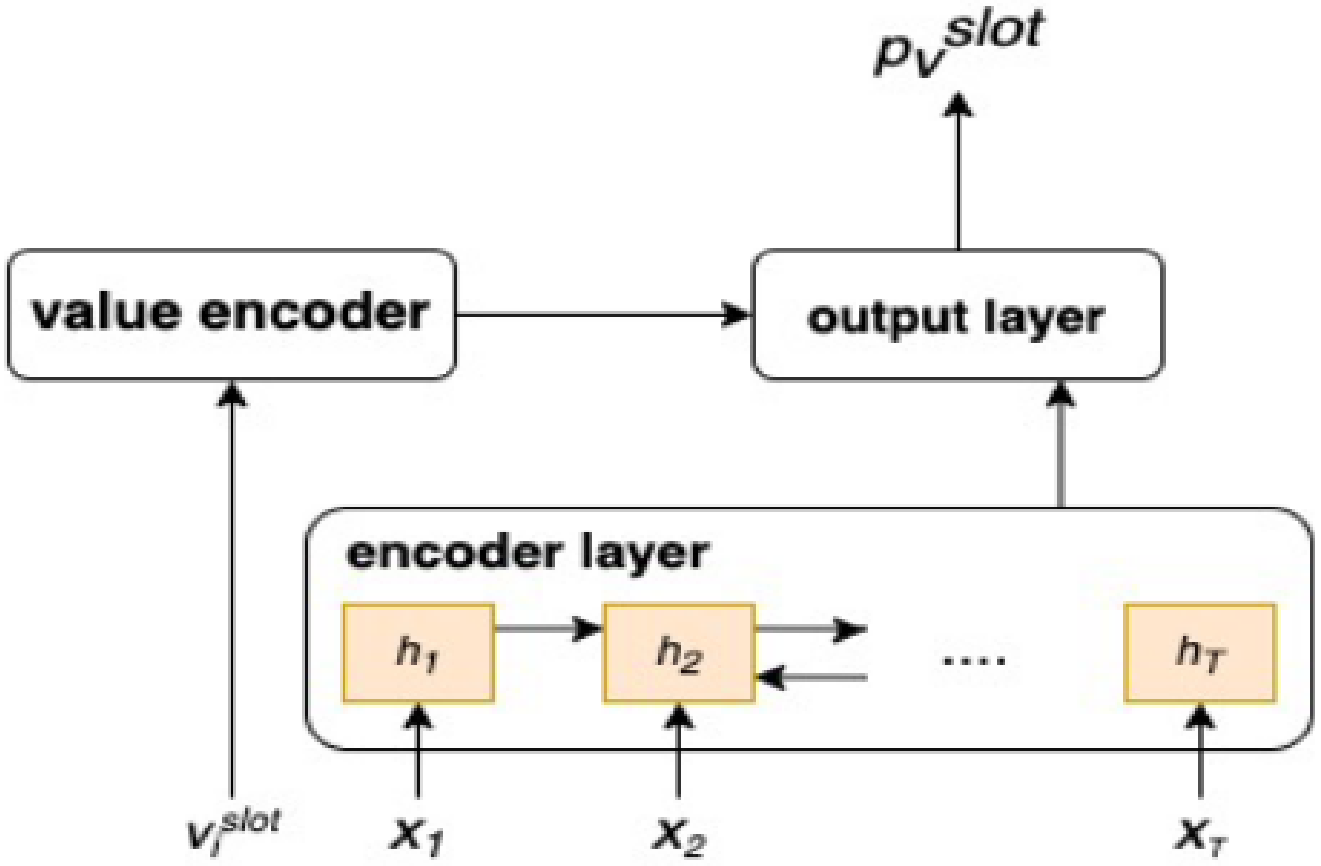
Static Ontology DST Models

Predict from a fixed set of slot-values

Advances

- Delexicalization
- Data-driven DST
- Parameter sharing
- Encoders based on pre-trained LM


y^{slot}
softmax
slot-specific output layer
encoder layer
h_1
h_2
....
h_T
x_1
x_2
x_T


p_V^{slot}
value encoder
output layer
encoder layer
h_1
h_2
....
h_T
v_i^{slot}
x_1
x_2
x_T

Delexicalization

e.g. *I want Chinese food* ➡ *I want F.VALUE F.SLOT*

- counter imbalanced training data for slot-values
- treats all slots using the same model parameters
  - allow transfer learning between slots
- relies on exact string matching to detect slot-value mentions
  - listing potential rephrasings for all slot values (semantic lexicon)
    ➡ not scalable

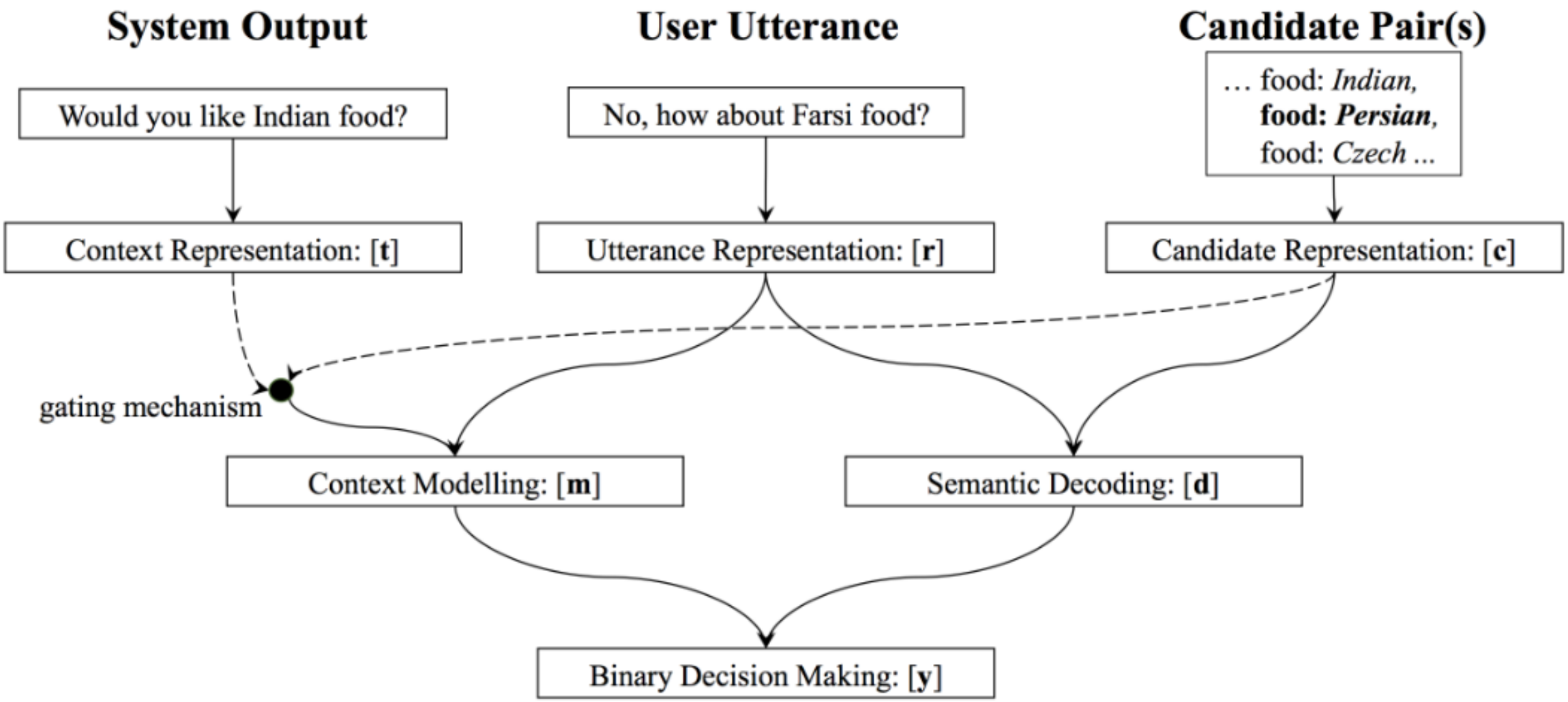
Robust dialog state tracking using delexicalised recurrent neural networks (Henderson et al., 2014)

Data-driven DST

Delexicalization requires additional manual feature engineering

Move past the word-based delexicalisation paradigm

- word vectors as sole building blocks for language understanding models



Neural belief tracker: Data-driven dialogue state tracking (Mrkšić et al., 2017)

Parameter Sharing

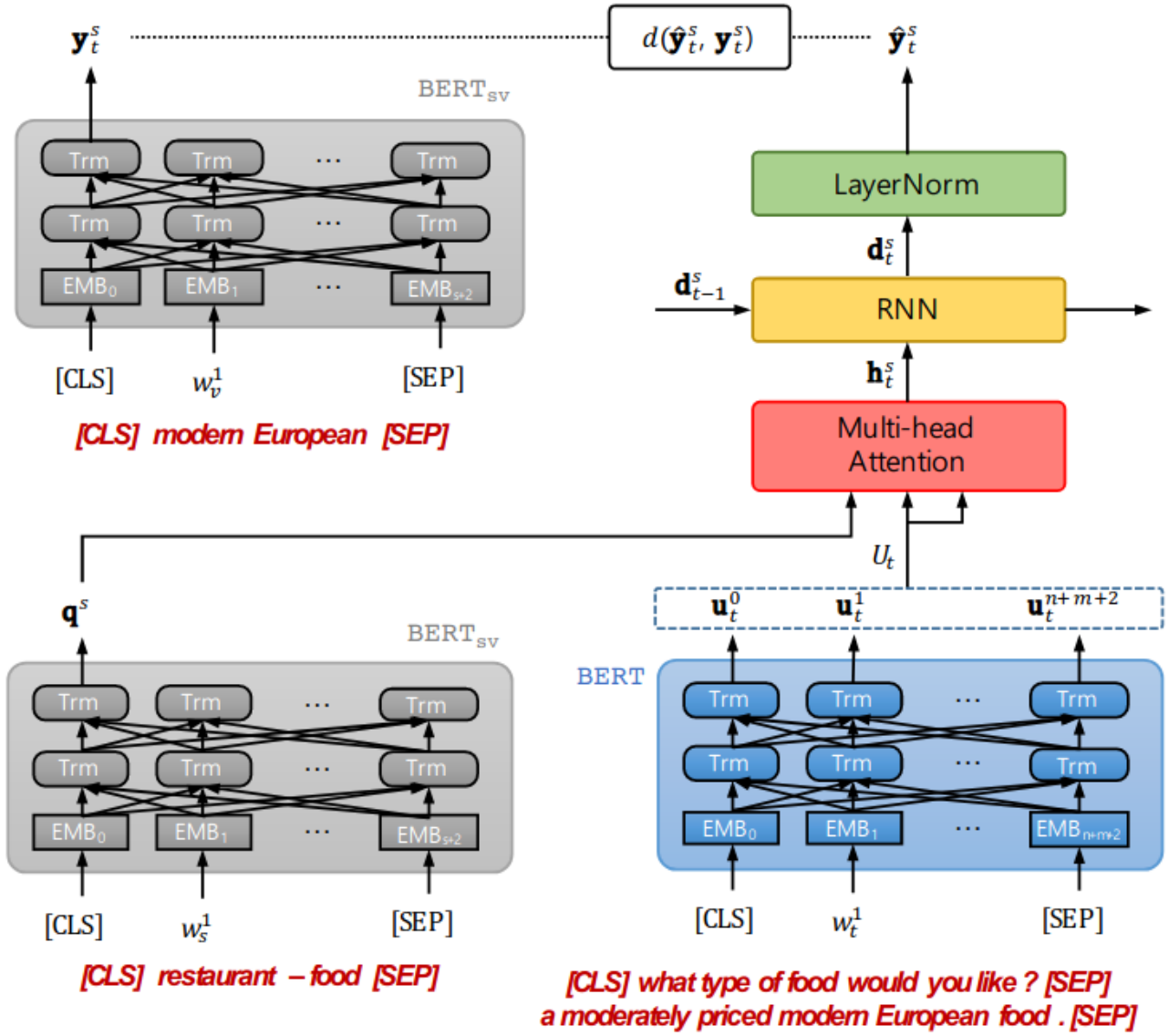
Neural belief tracker: separate encoder for each slot
➡ inefficient

StateNet, a DST model sharing the parameters for all slots

- combine an n-gram input feature representation with a slot representation
- LSTM to encode them into a single vector
- value representation compared with the encoded vector to obtain score for each slot-value

Encoders based on pre-trained LM

- using BERT to encode slots, user input, and slot-values
- input representation compared with slot-value representation



SUMBT: Slot-Utterance Matching for Universal and

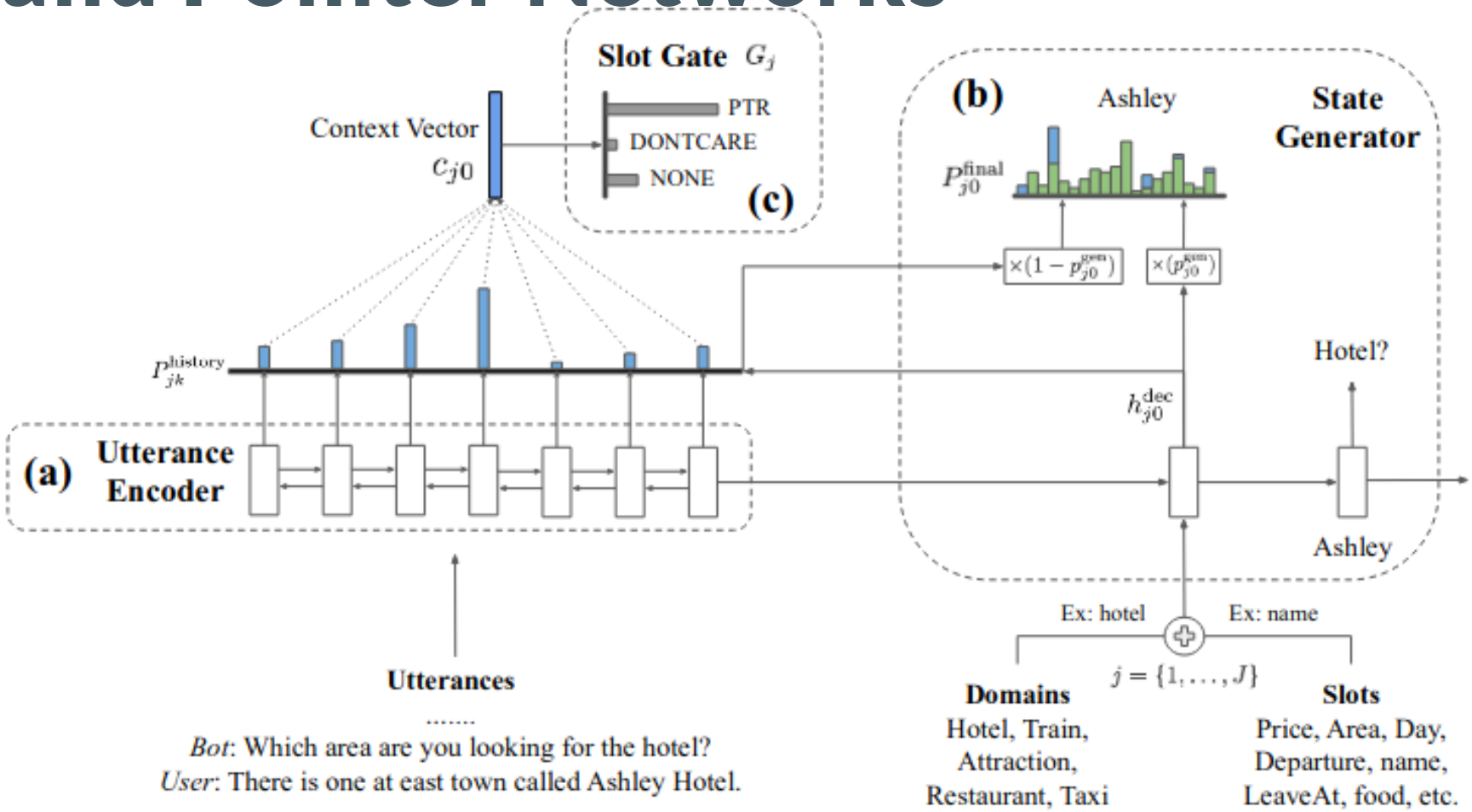
Dynamic Ontology DST Models

Predict from a possible open set of slot-values

Advances

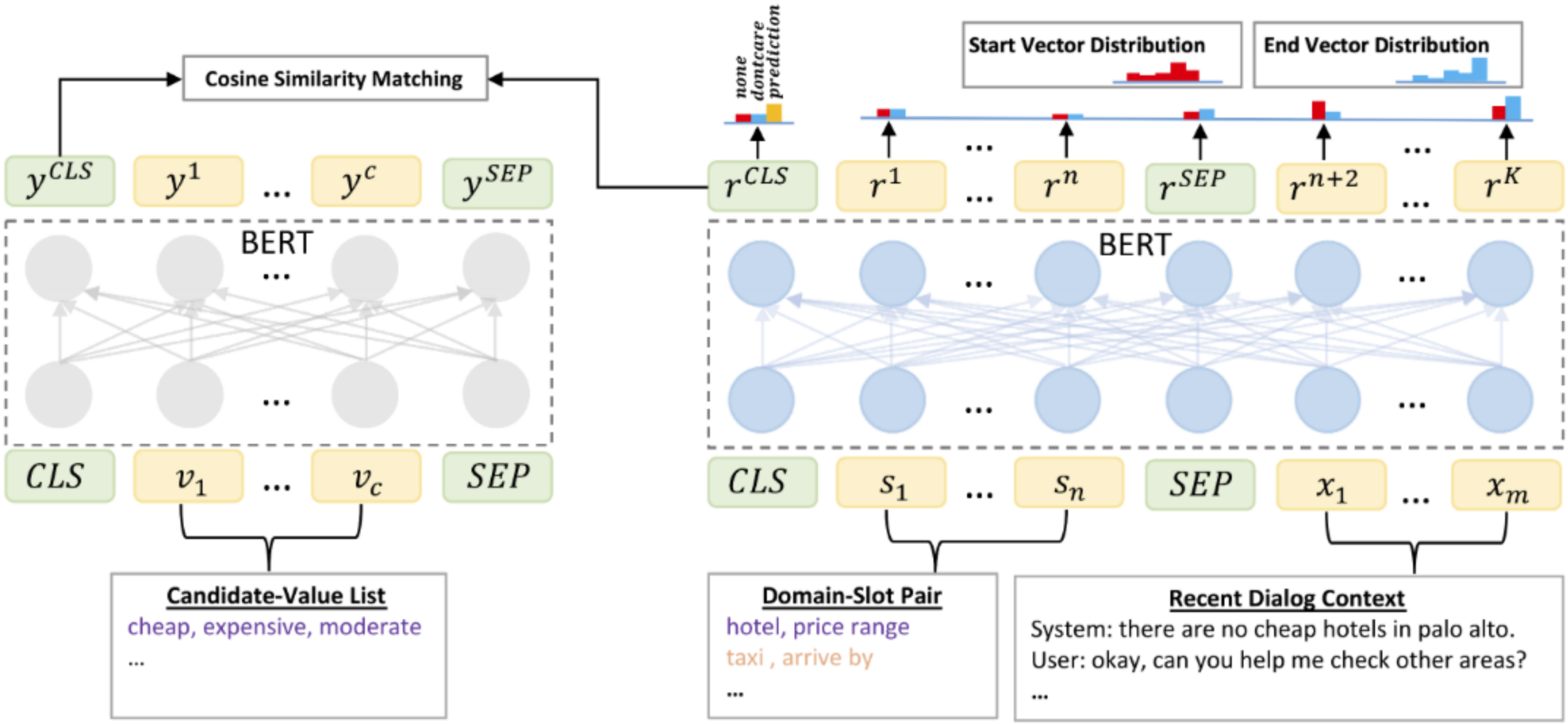
- Copy and pointer networks
- Categorical and non-categorical slot-values
- Schema-guided models
- Function-based update

Copy and Pointer Networks



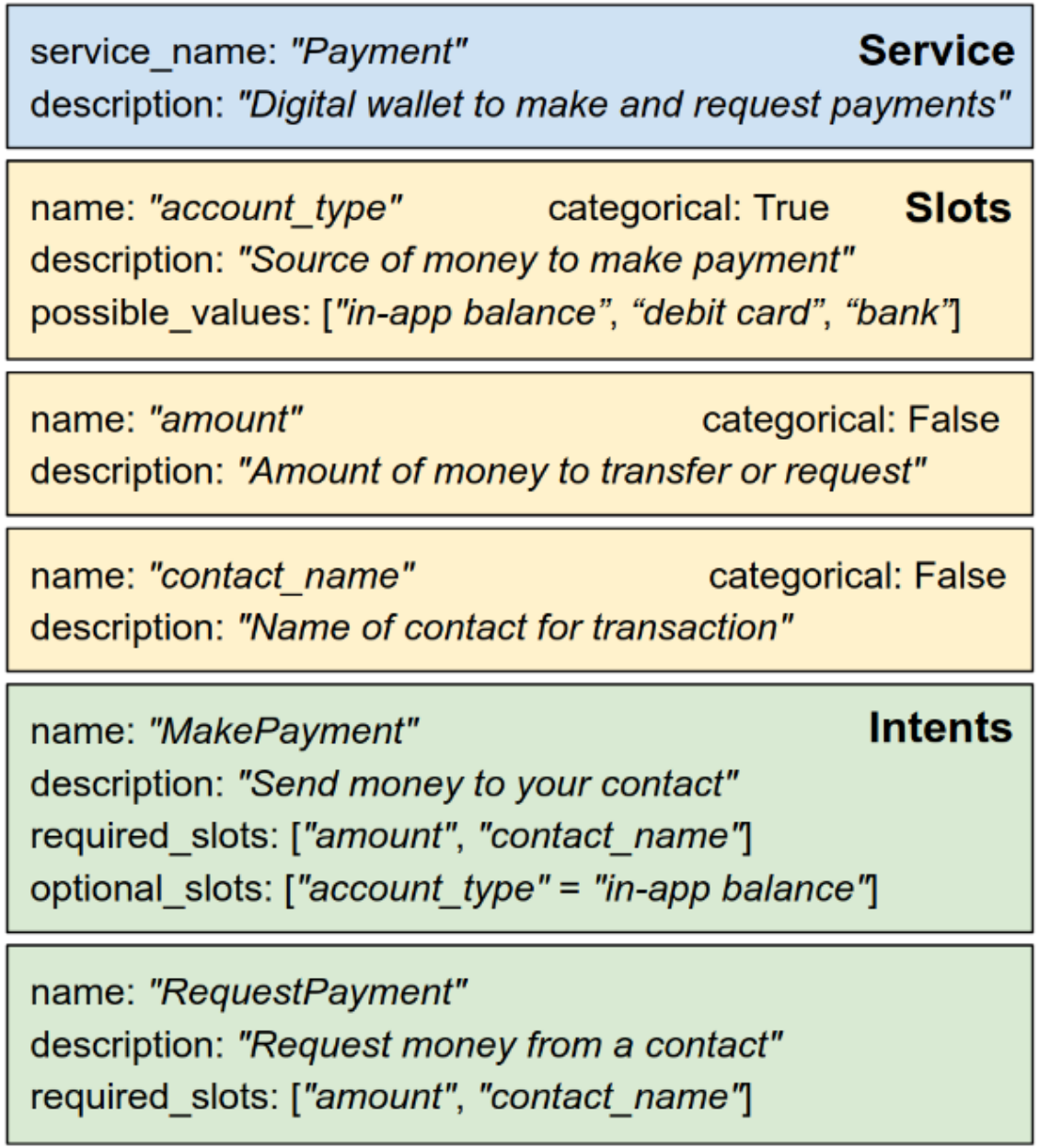
Transferable Multi-Domain State Generator (Wu et al., 2019)

Categorical and Non-Categorical Slot-Values

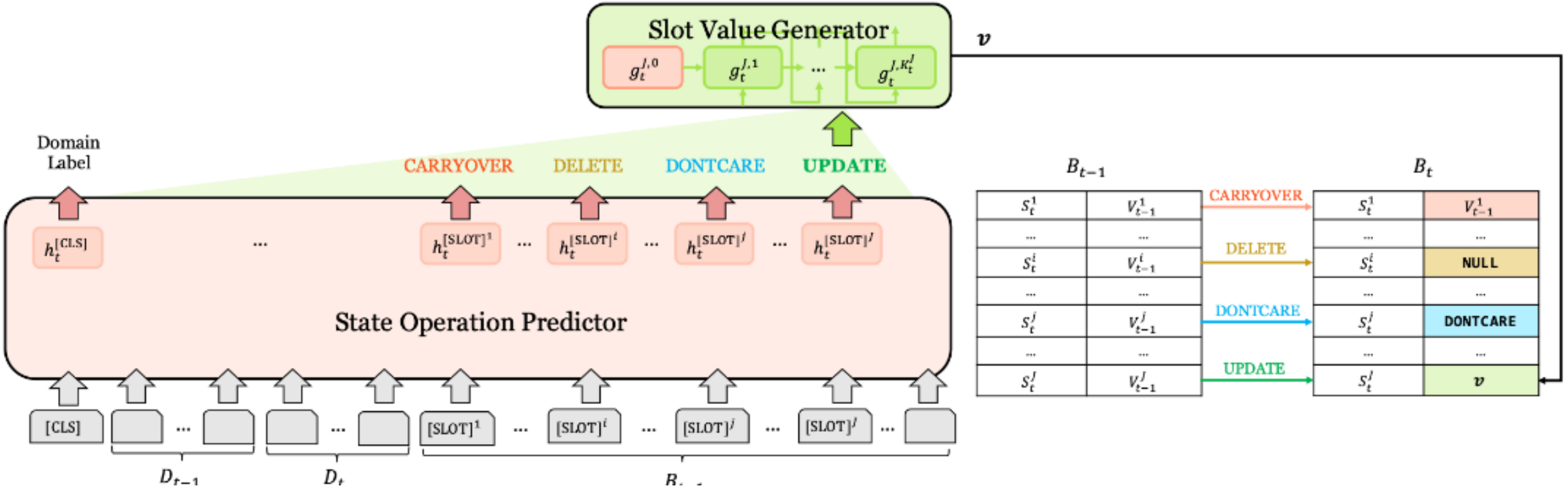


Schema-guided Models

- schema-guided dataset (Rastogi et al., 2020)
- standard schema to be adopted for all domains
- flexibility to incorporate new domains



Function-based Update



Challenges and Future Directions

Need for DST models that are

- **Generalizable**
  - to new domains / languages
  - without requiring new annotated data
- **Robust**
  - to longer dialogues and small variations in input
  - to challenges posed by spoken language
- **Efficient**
  - in representing and updating the dialogue state
  - resulting in better overall task-oriented dialogue performance

Generalizable

Domain Adaptation

Use of pre-trained models coupled with domain or task specialization

- task-adaptive pre-training
  - specialization corpora specific to task-oriented dialogues seem to be more useful than open-domain dialogues
- designing self-supervised objectives that can produce better representations of dialogues for the downstream task
  - e.g. response selection

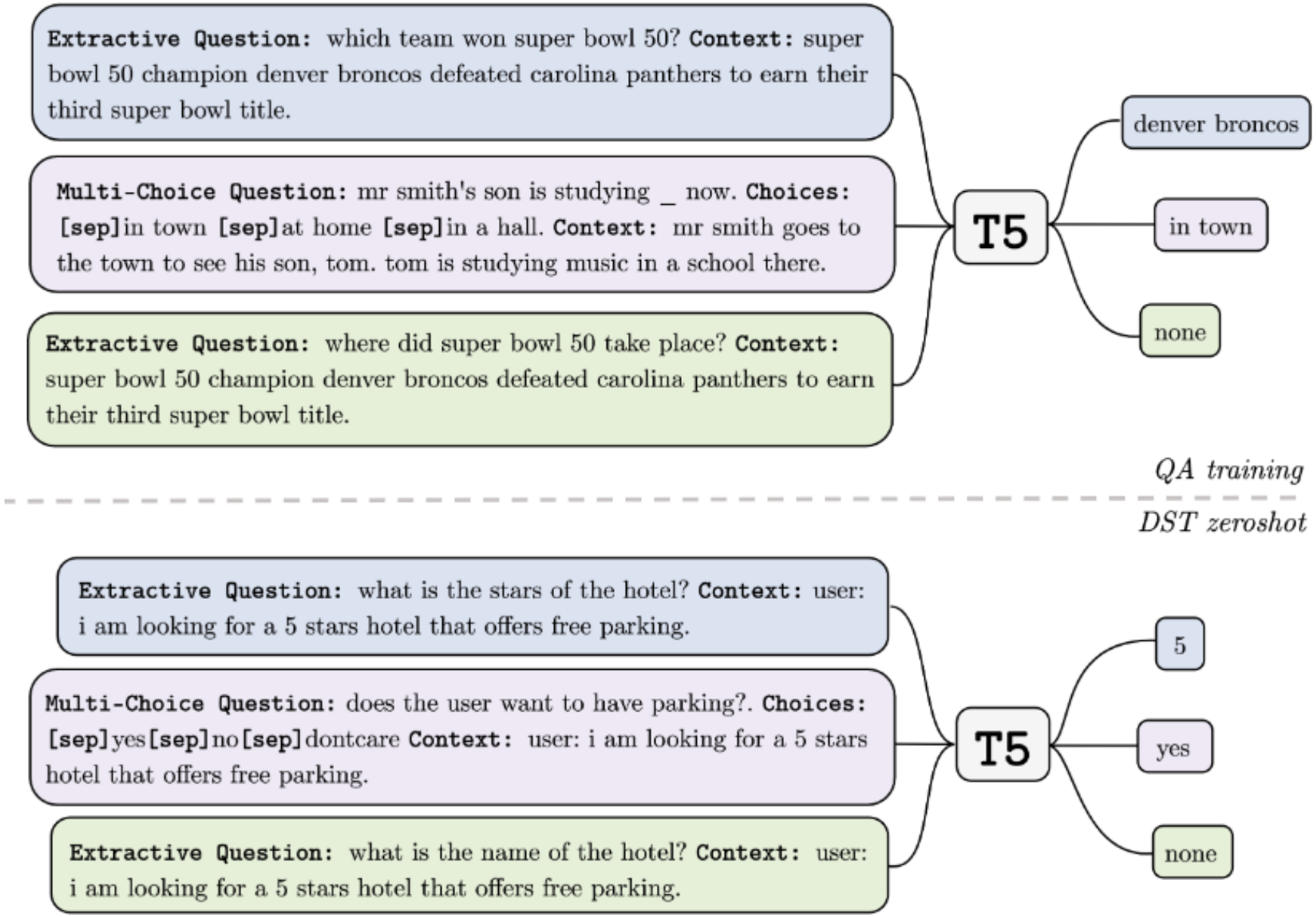
DS-TOD: Efficient Domain Specialization for Task Oriented Dialog (Hung et al., 2021)

Few-shot / Zero-shot Transfer Learning

Learn DST for a new domain or a new language given a much smaller in-domain corpus, or potentially no new annotated data

Two lines of work:

- cross-domain transfer learning (cross-lingual)
- cross-task transfer: leverage machine reading question answering (QA) data

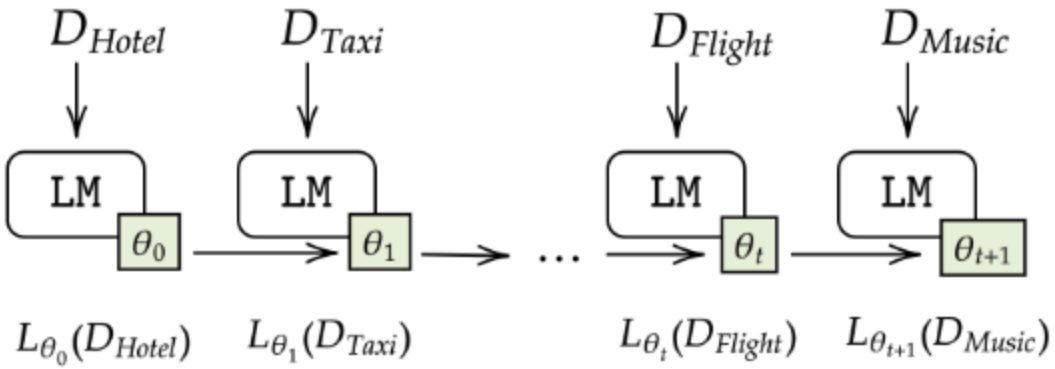

Extractive Question: which team won super bowl 50? Context: super bowl 50 champion denver broncos defeated carolina panthers to earn their third super bowl title.
Multi-Choice Question: mr smith's son is studying _ now. Choices: [sep]in town [sep]at home [sep]in a hall. Context: mr smith goes to the town to see his son, tom. tom is studying music in a school there.
Extractive Question: where did super bowl 50 take place? Context: super bowl 50 champion denver broncos defeated carolina panthers to earn their third super bowl title.
T5
denver broncos
in town
none
QA training
DST zeroshot
Extractive Question: what is the stars of the hotel? Context: user: i am looking for a 5 stars hotel that offers free parking.
Multi-Choice Question: does the user want to have parking?. Choices: [sep]yes[sep]no[sep]dontcare Context: user: i am looking for a 5 stars hotel that offers free parking.
Extractive Question: what is the name of the hotel? Context: user: i am looking for a 5 stars hotel that offers free parking.
T5
5
yes
none

Lifelong Learning

Learning new knowledge or skills through time sequentially

1. Adding slots to DST
2. Adding new domains

Without retraining with all data, the model should be able to accumulate knowledge



Robust

Long Dialogues

The longer a dialogue is, the less accurate dialogue state trackers are Can be attributed to:

- datasets containing longer term dependencies and spanning multiple contexts
- when the previous belief state is used to infer the current one: error propagation
- when the whole dialogue context is used to infer the current belief state: difficulty in distinguishing the relevant information in a longer context

DialogStitch: Synthetic Deeper and Multi-Context Task-Oriented Dialogs (Kottur et al. 2021)

Robust Models

To user inputs

- linguistic variation
- adversarial human attacks and uncooperative users, e.g. offensive behavior
- disfluencies and ungrammatical utterances (cfr. DATCHA corpus)

To speech inputs

- speech recognition errors
- data augmentation
- DSTC-10 Track 2 dataset: dev and test sets containing n-best ASR output (24.09% WER at 1-best)

➡ Need for diverse datasets that represent real-world challenges

Efficient

Beyond slot-filling: towards better dialogue state representations

Limitations of flat dialogue state representations:

- knowledge isn't directly shared across slots, e.g. a city can be an origin or a destination
- no handling of composition, e.g. "directions to my next meeting"

Alternatives exist

Graph

Dialog states as rooted relational graphs to encode compositionality

Hi can you book me a flight to Paris please.

```
user.flight.book.object.equals
  .destination.equals.location.equals.Paris
```

Sure, when and where will you depart?

```
system.prompt.flight.book.object.equals
  .source
  .departureDateTime
```

Queries

SQL query as a dialogue state

Q_2 : Which of those dorms have a TV lounge? INFORM_SQL

S_2 :
```
SELECT T1.dorm_name FROM dorm AS T1 JOIN has_amenity AS
T2 ON T1.dormid = T2.dormid JOIN dorm_amenity AS T3 ON
T2.amenid = T3.amenid WHERE T3.amenity_name = 'TV
Lounge'
```

A_2 : (Result table with many entries)

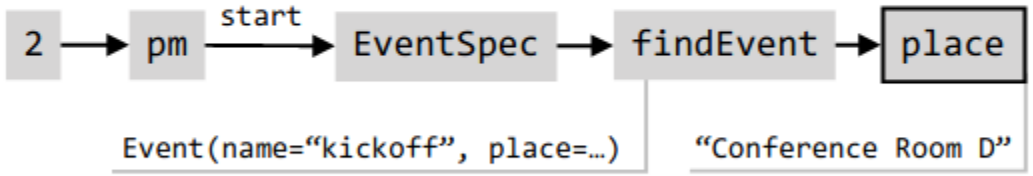
R_2 : This shows the names of dorms with TV lounges. CONFIRM_SQL

Data-flow

Executable program as state

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

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



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

Critical aspects of DST models

Slot status prediction

- crucial component of DST models
  - experiments with oracle slot status
- majority of slots are inactive resulting in a class imbalance problem

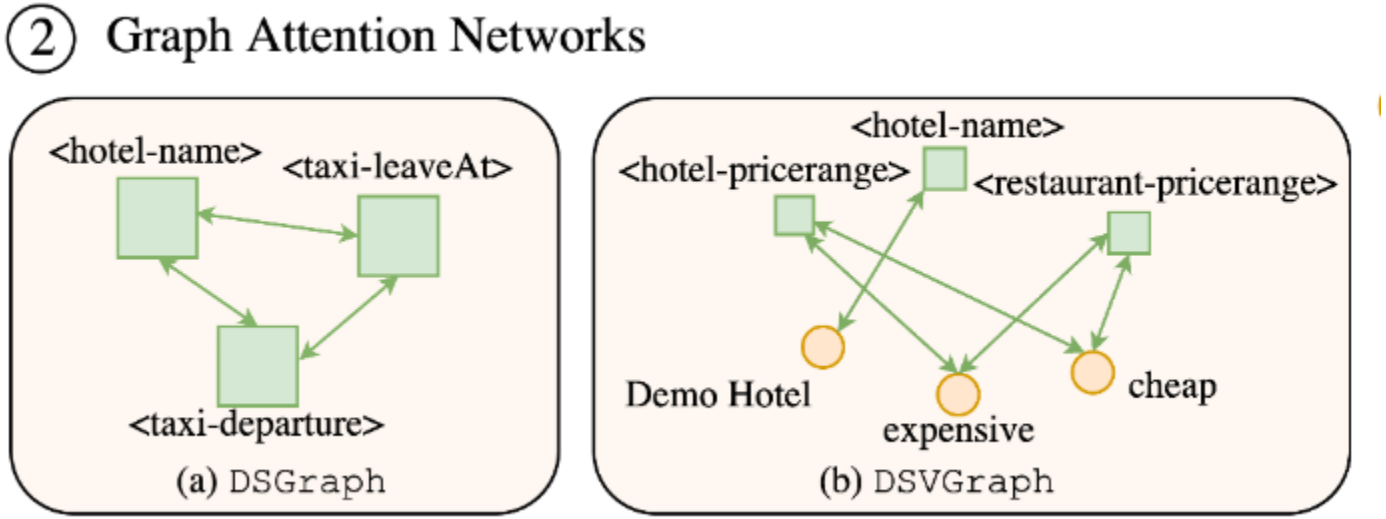
➡ Improving the *“none”* value slot accuracy has the potential to increase the overall DST performance by a large margin

Modelling inter-slot relationships

Slots are not conditionaly independent

- dependencies across domains

Using graph models to capture relationships between slots



Knowledge-Aware Graph-Enhanced GPT-2 for Dialogue State Tracking (Lin et al. 2021)

Dialogue state update

Potentially better ways than generating dialogue state from scratch or a simple rule-based update

- function-based update
  - SOM-DST: state operation to selectively update slot values at each turn
- prevent error propagation

Integrating the different dialogue modules

Each module optimized separately ➡ “credit assignment” problem

- does improved DST performance translate to better overall dialogue system performance (e.g. better task success rate)?
- end-to-end task-oriented dialogue
  - input: user utterance
  - output: system response

Thank you