Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures

Wenqiang Lei, Xisen Jin, Zhaochun Ren,
Xiangnan He, Min-Yen Kan, Dawei Yin

Traditional Pipeline Designs for Task-oriented Dialogue System

- Intent classifier
  - Booking restaurants etc.
- Belief tracker
- Policy maker
- Dialogue generator

Problems of Traditional Pipeline Designs

- Complex belief trackers
- Fragility
- Templated response

An End-to-end Solution

- Intent classifier
  - Booking restaurants etc.
- Belief tracker
- Policy maker
- Response generator

An **End-to-end** Trainable Dialogue System (NDM) (Wen et al., 2017b)

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, 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. EACL .

Some Problems Still Remains in NDM

- Complex belief trackers
  - Pre-trained Belief Tracker
- ~~Fragility~~
- ~~Templated response~~

Complex Belief Tracker In NDM

- Informable slots

Food style	Price range	Open hour	…
Chinese food	Expensive	Before 11:00 pm	…
Japanese food	Cheap	…	…
French food	…	…	…
…	…	…	…
…	…	…	…

- Requestable slots

Requiring address?	Requiring phone number?	Requiring name?	…
Yes	Yes	Yes	…
No	No	Know	…

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf> <Req>Address</Req>

Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf> <Req>Address</Req>

Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf> <Req>Address</Req>

Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf> <Req>Address</Req>

Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf>
    <Req>Address; Phone</Req>

Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Sequicity Solution

- Belief span
  - <Inf>Italian;Cheap</Inf> <Req>Address; Phone</Req>

- Notation
  - B_t: belief span
  - U_t: user utterance
  - R_t: machine response

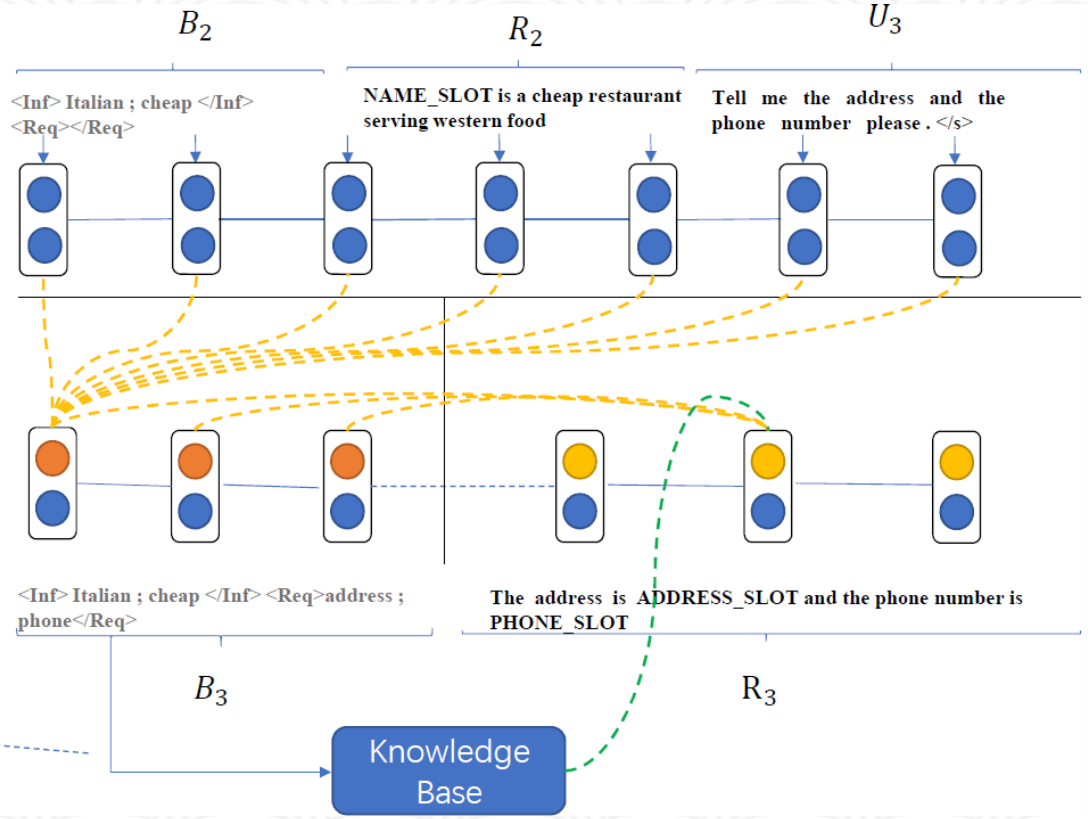
Turn	Dialogue
$User_1$	Can I have some Italian food please?
$Machine_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
$User_2$	I want cheap ones.
$Machine_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
$User_3$	Tell me the address and the phone number please .
$Machine_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT

Source sequence

Target sequence

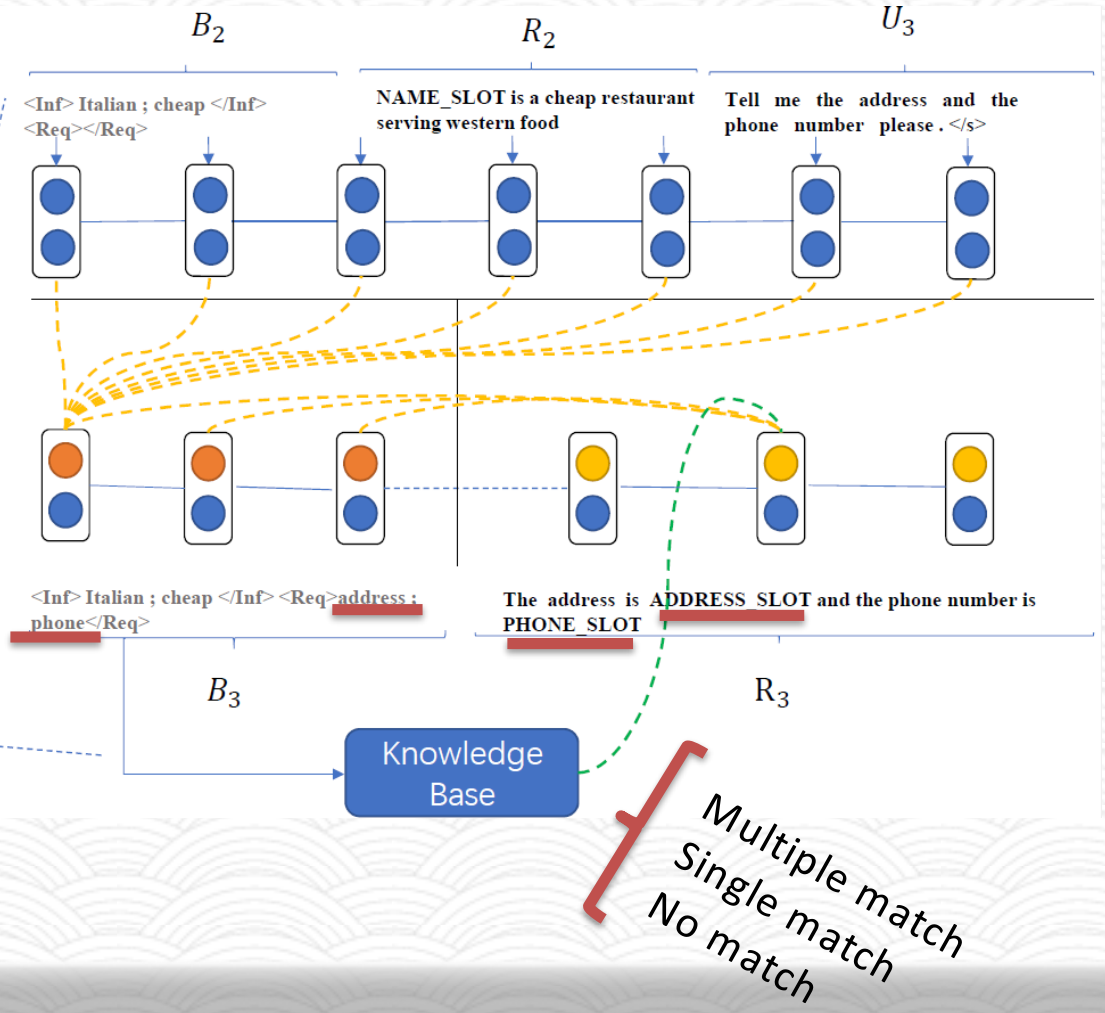
Sequicity Illustration

Turn	Dialogue
User$_1$	Can I have some Italian food please?
Machine$_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
User$_2$	I want cheap ones.
Machine$_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
User$_3$	Tell me the address and the phone number please .
Machine$_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT



Sequicity Illustration

Turn	Dialogue
User$_1$	Can I have some Italian food please?
Machine$_1$	<Inf> Italian </Inf><Req> </Req> What price range are you looking for?
User$_2$	I want cheap ones.
Machine$_2$	<Inf> Italian ; cheap </Inf> <Req></Req> NAME_SLOT is a cheap restaurant serving western food
User$_3$	Tell me the address and the phone number please .
Machine$_3$	<Inf> Italian ; cheap </Inf> <Req>address ; phone</Req> The address is ADDRESS_SLOT and the phone number is PHONE_SLOT



Optimization

- Joint log-likelihood
  - Short coming: treating each word equally
  - *E.g., The closest Italian restaurant is at* ***<addr_slot>***
- Reinforcement learning
  - Action: decoding a word
  - State: hidden vectors generated by RNNs
  - Reward: decoding a correct placeholder +1, decoding each word -0.1

Experiments: Datasets

Dataset	Cam676		
Size	Train:408 / Test: 136 / Dev: 136		
Domains	restaurant reservation		
Slot types	price, food style etc.		
Distinct slot values	99		

Dataset	KVRET		
Size	Train:2425 / Test: 302 / Dev: 302		
Domains	calendar	weather info.	POI
Slot types	date, etc.	location, etc.	poi, etc.
Distinct slot values	79	65	140

Experiment Results

	CamRes676					KVRET				
	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$
(1) NDM	0.904	0.212	0.832	91.9 min	8.6 min	0.724	0.186	0.741	285.5 min	29.3 min
(2) NDM + Att + SS	0.904	0.240	0.836	93.7 min	10.4 min	0.724	0.188	0.745	289.7 min	33.5 min
(3) LIDM	0.912	0.246	0.840	97.7 min	14.4 min	0.721	0.173	0.762	312.8 min	56.6 min
(4) KVRN	N/A	0.134	N/A	21.4 min	–	0.459	0.184	0.540	46.9 min	–
(5) TSCP	**0.927**	**0.253**	**0.854**	7.3 min	–	**0.845**	**0.219**	**0.811**	25.5 min	–
(6) Att-RNN	0.851	0.248	0.774	7.2 min	–	0.805	0.208	0.801	23.0 min	–
(7) TSCP\k_t	**0.927**	0.232	0.835	7.2 min	–	**0.845**	0.168	0.759	25.3 min	–
(8) TSCP\RL	**0.927**	0.234	0.834	**4.1** min	–	**0.845**	0.191	0.774	**17.5** min	–
(9) TSCP\B_t	0.888	0.197	0.809	22.9 min	–	0.628	0.182	0.755	42.7 min	–

Table 2: Model performance on CamRes676 and KVRET. This table is split into two parts: competitors on the upper side and our ablation study on the bottom side. **Mat.** and **Succ. F1** are for match rate and success F1 respectively. $\mathbf{Time_{full}}$ column reports training time till converge. For NDM, NDM+Att+SS and LIDM, we also calculate the training time for the rest parts except for the belief tracker ($\mathbf{Time_{N.B.}}$).

Time Expenses on Belief Trackers

	CamRes676					KVRET				
	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$
(1) NDM	0.904	0.212	0.832	91.9 min	8.6 min	0.724	0.186	0.741	285.5 min	29.3 min
(2) NDM + Att + SS	0.904	0.240	0.836	93.7 min	10.4 min	0.724	0.188	0.745	289.7 min	33.5 min
(3) LIDM	0.912	0.246	0.840	97.7 min	14.4 min	0.721	0.173	0.762	312.8 min	56.6 min
(4) KVRN	N/A	0.134	N/A	21.4 min	–	0.459	0.184	0.540	46.9 min	–
(5) TSCP	**0.927**	**0.253**	**0.854**	7.3 min	–	**0.845**	**0.219**	**0.811**	25.5 min	–
(6) Att-RNN	0.851	0.248	0.774	7.2 min	–	0.805	0.208	0.801	23.0 min	–
(7) TSCP\k_t	**0.927**	0.232	0.835	7.2 min	–	**0.845**	0.168	0.759	25.3 min	–
(8) TSCP\RL	**0.927**	0.234	0.834	**4.1** min	–	**0.845**	0.191	0.774	**17.5** min	–
(9) TSCP\B_t	0.888	0.197	0.809	22.9 min	–	0.628	0.182	0.755	42.7 min	–

Table 2: Model performance on CamRes676 and KVRET. This table is split into two parts: competitors on the upper side and our ablation study on the bottom side. **Mat.** and **Succ. F1** are for match rate and success F1 respectively. $\mathbf{Time_{full}}$ column reports training time till converge. For NDM, NDM+Att+SS and LIDM, we also calculate the training time for the rest parts except for the belief tracker ($\mathbf{Time_{N.B.}}$).

RL Helps with BLEU and Succ. F1

	CamRes676					KVRET				
	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$
(1) NDM	0.904	0.212	0.832	91.9 min	8.6 min	0.724	0.186	0.741	285.5 min	29.3 min
(2) NDM + Att + SS	0.904	0.240	0.836	93.7 min	10.4 min	0.724	0.188	0.745	289.7 min	33.5 min
(3) LIDM	0.912	0.246	0.840	97.7 min	14.4 min	0.721	0.173	0.762	312.8 min	56.6 min
(4) KVRN	N/A	0.134	N/A	21.4 min	–	0.459	0.184	0.540	46.9 min	–
(5) TSCP	**0.927**	**0.253**	**0.854**	7.3 min	–	**0.845**	**0.219**	**0.811**	25.5 min	–
(6) Att-RNN	0.851	0.248	0.774	7.2 min	–	0.805	0.208	0.801	23.0 min	–
(7) TSCP\k_t	**0.927**	0.232	0.835	7.2 min	–	**0.845**	0.168	0.759	25.3 min	–
(8) TSCP\RL	**0.927**	0.234	0.834	**4.1** min	–	**0.845**	0.191	0.774	**17.5** min	–
(9) TSCP\B_t	0.888	0.197	0.809	22.9 min	–	0.628	0.182	0.755	42.7 min	–

Table 2: Model performance on CamRes676 and KVRET. This table is split into two parts: competitors on the upper side and our ablation study on the bottom side. **Mat.** and **Succ. F1** are for match rate and success F1 respectively. $\mathbf{Time_{full}}$ column reports training time till converge. For NDM, NDM+Att+SS and LIDM, we also calculate the training time for the rest parts except for the belief tracker ($\mathbf{Time_{N.B.}}$).

Removing CopyNets

	CamRes676					KVRET				
	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$	Mat.	BLEU	Succ. F_1	$Time_{full}$	$Time_{N.B.}$
(1) NDM	0.904	0.212	0.832	91.9 min	8.6 min	0.724	0.186	0.741	285.5 min	29.3 min
(2) NDM + Att + SS	0.904	0.240	0.836	93.7 min	10.4 min	0.724	0.188	0.745	289.7 min	33.5 min
(3) LIDM	0.912	0.246	0.840	97.7 min	14.4 min	0.721	0.173	0.762	312.8 min	56.6 min
(4) KVRN	N/A	0.134	N/A	21.4 min	–	0.459	0.184	0.540	46.9 min	–
(5) TSCP	**0.927**	**0.253**	**0.854**	7.3 min	–	**0.845**	**0.219**	**0.811**	25.5 min	–
(6) Att-RNN	0.851	0.248	0.774	7.2 min	–	0.805	0.208	0.801	23.0 min	–
(7) TSCP\k_t	**0.927**	0.232	0.835	7.2 min	–	**0.845**	0.168	0.759	25.3 min	–
(8) TSCP\RL	**0.927**	0.234	0.834	**4.1** min	–	**0.845**	0.191	0.774	**17.5** min	–
(9) TSCP\B_t	0.888	0.197	0.809	22.9 min	–	0.628	0.182	0.755	42.7 min	–

Table 2: Model performance on CamRes676 and KVRET. This table is split into two parts: competitors on the upper side and our ablation study on the bottom side. **Mat.** and **Succ. F1** are for match rate and success F1 respectively. $\mathbf{Time_{full}}$ column reports training time till converge. For NDM, NDM+Att+SS and LIDM, we also calculate the training time for the rest parts except for the belief tracker ($\mathbf{Time_{N.B.}}$).

Discussions: OOV Experiments

Synthesized OOV data:
I would like some **Chinese** food. → I would like some **Chinese_unk** food.

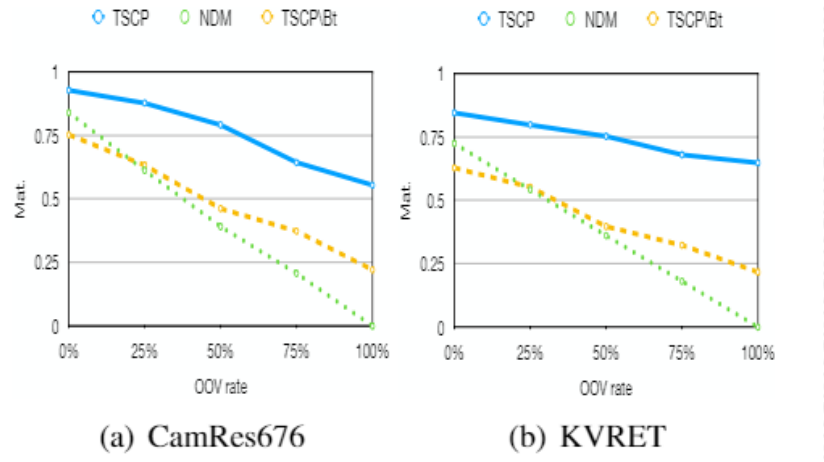


Figure 2: OOV tests. 0% OOV rate means no OOV instance while 100% OOV rate means all instances are changed to be OOV.

Discussion: Parameter Scales

Dataset	Cam676		
Size	Train:408 / Test: 136 / Dev: 136		
Domains	restaurant reservation		
Slot types	price, food style etc.		
Distinct slot values	99		

Dataset	KVRET		
Size	Train:2425 / Test: 302 / Dev: 302		
Domains	calendar	weather info.	POI
Slot types	date, etc.	location, etc.	poi, etc.
Distinct slot values	79	65	140

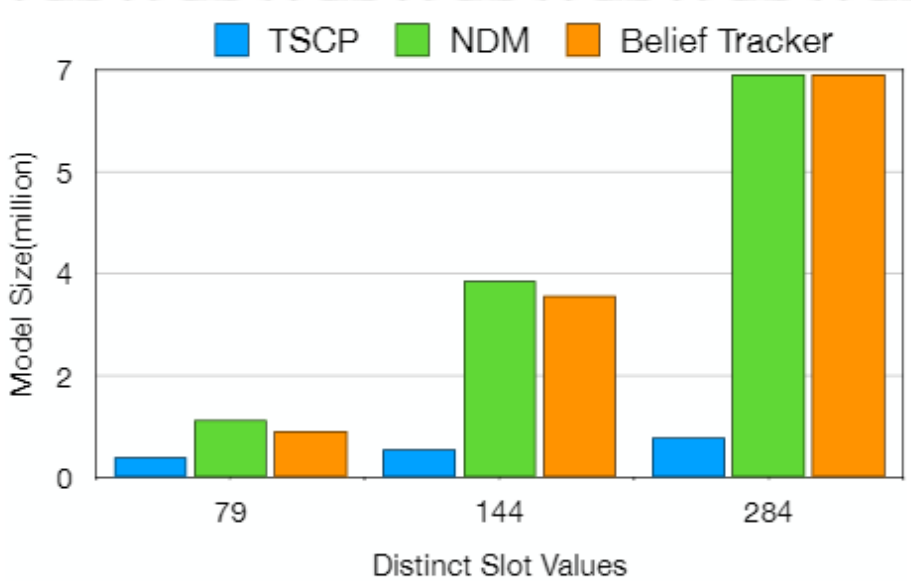


Figure 3: Model size sensitivity with respect to KVRET. Distinct slot values of 79, 144, 284 correspond to the number of slots in KVRET's *calendar*, *calendar* + *weather info.*, and all 3 domains.

Discussion: Parameter Scales

Dataset	Cam676		
Size	Train:408 / Test: 136 / Dev: 136		
Domains	restaurant reservation		
Slot types	price, food style etc.		
Distinct slot values	99		

Dataset	KVRET		
Size	Train:2425 / Test: 302 / Dev: 302		
Domains	calendar	weather info.	POI
Slot types	date, etc.	location, etc.	poi, etc.
Distinct slot values	79	65	140

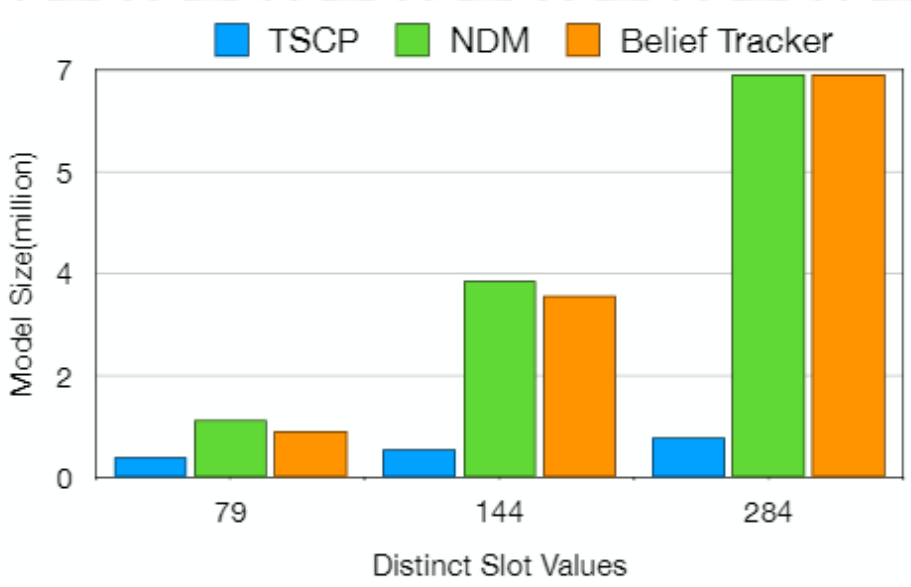


Figure 3: Model size sensitivity with respect to KVRET. Distinct slot values of 79, 144, 284 correspond to the number of slots in KVRET's *calendar*, *calendar* + *weather info.*, and all 3 domains.

Conclusion

- Sequicity provides another direction for task-oriented dialogue systems.
- It is more light-weighted, can handle OOV requests.
- It learns dialogue action directly from data with less human interventions
  - Requires more training data.