TRADE: Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems

Chien-Sheng (Jason) Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher and Pascale Fung

2019.07.29

https://github.com/jasonwu0731/trade-dst
Short Conclusion

TRADE is a simple copy-augmented generative model that can track dialogue states without requiring ontology. It is the current SOTA model in multi-domain DST. It also enables zero-shot and few-shot DST in an unseen domain.
Dialogue Systems: Chit-Chat v.s. Task-Oriented

Chit-Chat Dialogue Systems
▷ No Specific goal
▷ Focus on generating natural responses
▷ The more turns the better

Task-Oriented Dialogue Systems
▷ Help users achieve their goal
▷ Focus on understanding users, tracking states, and generating next actions.
▷ The less turns the better
Modularized Dialogue Systems

- Domain Identification
- Intent Detection
- Slot Filling

Represent system’s belief of user’s goal as slot-value pairs

Mapping actions and states into natural language.

Query knowledge base
Dialogue policy (generate next action)
Problem: Multi-Domain Dialogue State Tracking (DST)

Multi-domain, Multi-turn Conversations

Machine Learning Models

Dialogue States
System belief of a user’s goal or Intention.
Ex: Restaurant:
{Location: Florence, Price: Expensive, ...}
A Dialogue Example

I’m looking for a cheap pizza restaurant in the city center.

Sure. There is a Dominos nearby. How many?

Four people Tuesday at 1 pm please. Please make sure there is NO PINEAPPLE on the pizza!

Booked! ABCDE is your reservation code.

Also looking for some architectural place close to restaurant.

Florence Cathedral is famous. Would you like to head there?

Yes help me book a taxi between restaurant and church.

What time do you need the taxi?

Around 3pm please.

You are all set. Have a good trip.
Ontology-based DST

- Given system response and current user utterance, each slot in each domain is predicted to be one of the predefined values in ontology.
- Models: ScaleDST (Rastogi et al., 2017); MDBT (Ramadan et al., 2018); GLAD (Zhong et al., 2018); GCE (Nouri et al., 2018)
Challenges of Ontology-based DST

- Ontology is hard to obtain in real scenarios
- Need to track lots of slot values
- Cannot track unseen slot values
- Missing domain sharing capacities

Find a train at 5pm

Find a train at 5pm

Find a train at 5:05

Find a taxi at 5pm

| Type: A, B, C, D, E, F, G, H, I, J, K, L, ... |
| Leave at: ? |
| Leave at: 5pm |
| Leave at: ? |
| Leave at: ? |
Usr: Find me a cheap restaurant at 7 pm.
Sys: What cuisine would you like?
Usr: I’d prefer eating sushi or ramen.
Sys: Where should it be?
Usr: Let’s do in Florence. Also I need a taxi to go there at 6:30 pm.
Sys: ...?
Usr: Find me a cheap restaurant at 7 pm.
Sys: What cuisine would you like?
Usr: I’d prefer eating sushi or ramen.
Sys: Where should it be?
Usr: Let’s do it in Florence. Also I need a taxi to go there at 6:30 pm.
Sys: …?
Usr: Find me a cheap restaurant at 7 pm.
Sys: What cuisine would you like?
Usr: I’d prefer eating sushi or ramen.
Sys: Where should it be?
Usr: Let’s do in Florence. Also I need a taxi to go there at 6:30 pm.
Sys: …?
Usr: Find me a cheap restaurant at 7 pm.
Sys: What cuisine would you like?
Usr: I’d prefer eating sushi or ramen.
Sys: Where should it be?
Usr: Let’s do in Florence. Also I need a taxi to go there at 6:30 pm.
Sys: ...?
Usr: Find me a cheap restaurant at 7 pm.
Sys: What cuisine would you like?
Usr: I’d prefer eating sushi or ramen.
Sys: Where should it be?
Usr: Let’s do in Florence. Also I need a taxi to go there at 6:30 pm.
Sys: ...?
Sequence-to-Sequence (Seq2Seq)

Encoder RNN

Word Embeddings

Decoder RNN

Vocabulary Distribution

$p_{vocab} = \text{Softmax}(E(h_{jk}^{dec})^T) \in \mathbb{R}^{|V|}$
Seq2Seq with Attention

\[ p_{\text{history}} = \text{Softmax}(H_t(h_{jk}^{\text{dec}})^T) \in \mathbb{R}^{|X_t|}. \]
Seq2Seq with (Soft) Copy Mechanism

(See et al. 2017)

Encoder RNN

Word Embeddings

Context Vector

Attention Weights

Decoder RNN

$P_{jk}^{gen} = \text{Sigmoid}(W_{1}[h_{jk}^{dec}; w_{jk}; c_{jk}]) \in \mathbb{R}^1$, 

$x_p$

$x(1-p)$
**TRADE**: Transferable Dialogue State Generator

---

(2) **Slot Gate**

Context Vector $c_{j0}$

$G_j = \text{Softmax}(W_g \cdot (c_{j0})^T) \in \mathbb{R}^3$,

---

(1) **Utterance Encoder**

Utterances

*Bot*: Which area are you looking for the hotel?

*User*: There is one at east town called Ashley Hotel.

---

(b) **State Generator**

Ex: hotel

State Generator

---

(c) **State Generator**

Ex: name

---

**Domains**

Hotel, Train, Attraction, Restaurant, Taxi

---

**Slots**

Price, Area, Day, Departure, name, LeaveAt, food, etc.
**TRADE:** Transferable Dialogue State Generator

**Utterances**

*Bot:* Which area are you looking for the hotel?
*User:* There is one at east town called Ashley Hotel.

**Domains**

Hotel, Train, Attraction, Restaurant, Taxi

**Slots**

Price, Area, Day, Departure, name, LeaveAt, food, etc.

**Slot Gate**

\[ L_g = \sum_{j=1}^{J} - \log(G_j \cdot (y_{j}^{gate})^\top) \]

**State Generator**

\[ L_u = \sum_{j=1}^{J} \sum_{k=1}^{|Y_j|} - \log(P_{jk}^{\text{final}} \cdot (y_{jk}^{\text{value}})^\top) \]

**Ex:** hotel

**Ex:** Ashley

**Don't Care**

*PTR*

*NONE*
MultiWOZ Dataset (Budzianowski et al., 2018)

- The largest available human-human conversational corpus with DST labels (8438 dialogues with avg 13.68 turns).
- 5 domains (Hotel, Train, Attraction, Restaurant, Taxi) and 16 slots (food, leave at, area, etc).
- Total 30 domain-slot pairs and ~4500 slot values.
Multi-Domain Joint Training

- MDBT (Ramadan et al., 2018)
- GLAD (Zhong et al., 2018)
- GCE (Nouri et al., 2018)
- SpanPtr (Xu et al., 2018)
Multi-Domain Joint Training: Visualization
Zero-Shot Domain DST

<table>
<thead>
<tr>
<th></th>
<th>Trained Single</th>
<th>Zero-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint</td>
<td>Slot</td>
</tr>
<tr>
<td>Hotel</td>
<td>55.52</td>
<td>92.66</td>
</tr>
<tr>
<td>Train</td>
<td>77.71</td>
<td>95.30</td>
</tr>
<tr>
<td>Attraction</td>
<td>71.64</td>
<td>88.97</td>
</tr>
<tr>
<td>Restaurant</td>
<td>65.35</td>
<td>93.28</td>
</tr>
<tr>
<td>Taxi</td>
<td>76.13</td>
<td>89.53</td>
</tr>
</tbody>
</table>

Table 3: Zero-shot experiments on an unseen domain. In taxi domain, our model achieves 60.58% joint goal accuracy without training on any samples from taxi domain. Trained Single column is the results achieved by training on 100% single-domain data as a reference.
Unseen Domain Testing (Zero-Shot): Correctness Analysis

Hotel

Restaurant
Few-Shot Domain Expansion DST: (1% unseen domain data)

▷ Why?
  ○ Be able to quickly adapt to new domains.
  ○ Not require retraining with all the data from previously learned domains (not available and time-consuming).

▷ How?
  ○ Naive fine-tuning; EWC (Kirkpatrick et al., 2017); GEM (Lopez-Paz et al., 2017).

▷ What?
  ○ Unseen domain performance
  ○ Trained domains performance

\[
L_{ewc}(\Theta) = L(\Theta) + \sum_i \frac{\lambda}{2} F_i(\Theta_i - \Theta_{S,i})^2
\]

Minimize \( L(\Theta) \)
Subject to \( L(\Theta, K) \leq L(\Theta_S, K) \),

24
Unseen Domain Performance (Few-Shot)
Unseen Domain Performance (Few-Shot)
Trained Domains Performance (Few-Shot)
MultiWOZ 2.1 (Eric et al., 2019)

A correction version of original MultiWOZ dataset, resulting in changes to 32% of state annotations across 40% of the dialogue turns.

<table>
<thead>
<tr>
<th>Model</th>
<th>MultiWOZ 2.0</th>
<th>MultiWOZ 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FJST</td>
<td>40.2%</td>
<td>38.0%</td>
</tr>
<tr>
<td>HJST</td>
<td>38.4%</td>
<td>35.55%</td>
</tr>
<tr>
<td>TRADE</td>
<td>48.6%</td>
<td>45.6%</td>
</tr>
<tr>
<td>DST Reader</td>
<td>39.41%</td>
<td>36.4%</td>
</tr>
<tr>
<td>HyST</td>
<td>42.33%</td>
<td>38.1%</td>
</tr>
</tbody>
</table>

Table 5: Examples of annotation errors between MultiWOZ 2.0 and 2.1
Thank you!
Any Questions?

Chien-Sheng (Jason) Wu
Andrea Madotto
Ehsan Hosseini-Asl

Caiming Xiong
Richard Socher
Pascale Fung

Code: [https://github.com/jasonwu0731/trade-dst](https://github.com/jasonwu0731/trade-dst)