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.
Hey, Google, turn off the kitchen (light)!
(PLEASE TYPE IN ALL CAPS) WHAT'S YOUR NAME DEAR?

HI, I'M ELIZA. WHAT DO YOU WANT TO TALK ABOUT?
CAN I TALK ABOUT MY PROBLEM?
SURE...
DUE TO ONGOING EXAMS IN COLLEGE, I HAVE A LOT OF STRESS.
PLEASE RELAX A LITTLE. SLEEP WELL.
THANKS FOR YOUR ADVICE.
NO MENTION.
BYE
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
Task-Oriented Dialogue System

Input Speech

Automatic Speech Recognition (ASR) → Spoken Language Understanding (SLU) → Dialog State Tracking (DST) → Belief State

Text to Speech (TTS) → Natural Language Generation (NLG) → System Response

Voice Output

Speech Recognition (ASR) → Domain/Intents/Slot Values

Text in

Text out
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
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
## Database Example

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

...
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

- **Automatic Speech Recognition (ASR)**
- **Spoken Language Understanding (SLU)**
- **Dialog State Tracking (DST)**
- **Text To Speech (TTS)**
- **Natural Language Generation (NLG)**
- **Dialog Policy**

Flowchart showing the interaction between input speech, user actions, and system responses.
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 Artificial Intelligence 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.

<table>
<thead>
<tr>
<th>user msg</th>
<th>Hi! I’m looking for good thriller. Are there any playing right now?</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent msg</td>
<td>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?</td>
</tr>
<tr>
<td>agent acts</td>
<td>inform(moviename=The Witch, The Other Side of the Door, The Boy; genre=thriller) multiple_choice(moviename)</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>annotation</th>
<th>inform(moviename=The Witch, The Other Side of the Door, The Boy; genre=thriller) multiple_choice(moviename)</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification</td>
<td>inform+moviename, inform+genre, multiple_choice+moviename</td>
</tr>
<tr>
<td>sequence</td>
<td>‘inform’ ‘(‘moviename’ ‘=’ ‘;’ ‘genre’ ‘=’ ‘)’ ‘multiple_choice’ ‘(‘moviename’ ‘)’ ‘(eos)’</td>
</tr>
<tr>
<td>cas sequence</td>
<td>⟨⟨continue⟩, inform, {moviename, genre}⟩ ⟨⟨continue⟩, multiple_choice, {moviename}⟩ ⟨⟨stop⟩, ⟨pad⟩, {}⟩</td>
</tr>
</tbody>
</table>
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)
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> \}

\[
x_t^c = W_x^c [c_{t-1}, a_{t-1}, s_{t-1}, k] + b_x^c,
\]

\[
g_t^c , h_t^c = \text{GRU}^c (x_t^c, h_{t-1}),
\]

\[
P(c_t) = \text{softmax}(W_g^c g_t^c + b_g^c),
\]

\[
\mathcal{L}^c = - \sum_t \log P(c_t).
\]
act unit

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

**Output:** one act from act space

\[
x_t^a = W_x^a [c_t, a_{t-1}, s_{t-1}, k] + b_x^a,
\]

\[
g_t^a, h_t^a = \text{GRU}^a (x_t^a, h_{t-1}^c),
\]

\[
P(a_t) = \text{softmax}(W_g^a g_t^a + b_g^a),
\]

\[
\mathcal{L}^a = - \sum_t \log P(a_t).
\]
**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.

\[
x_t^s = W_x^s[c_t, a_t, s_{t-1}, k] + b_x^s,
\]

\[
g_t^s, h_t^s = \text{GRU}^s(x_t^s, h_t^a),
\]

\[
s_t = \text{sigmoid}(W_g^s g_t^s + b_g^s),
\]

\[
\mathcal{L}^s = - \sum_t \sum_{i=0}^{\vert S \vert} z_t^i \log s_t^i + (1 - z_t^i) \log (1 - s_t^i).
\]

**Overall Loss**

\[
\mathcal{L} = \mathcal{L}_c^a + \mathcal{L}^a + \mathcal{L}^s
\]
## Dataset

Microsoft Research End-to-End Dialogue Challenge

<table>
<thead>
<tr>
<th>domain</th>
<th>total</th>
<th>train</th>
<th>valid</th>
<th>test</th>
<th>acts</th>
<th>slots</th>
<th>pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>movie</td>
<td>2888</td>
<td>1445</td>
<td>433</td>
<td>1010</td>
<td>11</td>
<td>29</td>
<td>90</td>
</tr>
<tr>
<td>taxi</td>
<td>3093</td>
<td>1548</td>
<td>463</td>
<td>1082</td>
<td>11</td>
<td>23</td>
<td>63</td>
</tr>
<tr>
<td>restaurant</td>
<td>4101</td>
<td>2051</td>
<td>615</td>
<td>1435</td>
<td>11</td>
<td>31</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>domain &amp; speaker</th>
<th>1 act</th>
<th>2 acts</th>
<th>3 acts</th>
<th>4 acts</th>
</tr>
</thead>
<tbody>
<tr>
<td>movie user</td>
<td>9130</td>
<td>1275</td>
<td>106</td>
<td>11</td>
</tr>
<tr>
<td>movie agent</td>
<td>5078</td>
<td>4982</td>
<td>427</td>
<td>33</td>
</tr>
<tr>
<td>taxi user</td>
<td>10544</td>
<td>762</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>taxi agent</td>
<td>7855</td>
<td>3301</td>
<td>200</td>
<td>8</td>
</tr>
<tr>
<td>restaurant user</td>
<td>12726</td>
<td>1672</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>restaurant agent</td>
<td>10333</td>
<td>3755</td>
<td>403</td>
<td>10</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Task</th>
<th>Entity F&lt;sub&gt;1&lt;/sub&gt;</th>
<th>Success F&lt;sub&gt;1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>movie</td>
<td>taxi</td>
</tr>
<tr>
<td>Classification</td>
<td>34.02</td>
<td>49.71</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>39.95</td>
<td>63.12</td>
</tr>
<tr>
<td>Copy Seq2Seq</td>
<td>28.04</td>
<td>62.95</td>
</tr>
<tr>
<td>CAS</td>
<td>48.02</td>
<td>59.16</td>
</tr>
<tr>
<td>gCAS</td>
<td><strong>50.86</strong></td>
<td><strong>64.00</strong></td>
</tr>
<tr>
<td>method</td>
<td>movie</td>
<td>taxi</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{R}$</td>
</tr>
<tr>
<td>classification</td>
<td>84.19</td>
<td>50.24</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>73.44</td>
<td>73.62</td>
</tr>
<tr>
<td>Copy Seq2Seq</td>
<td>67.56</td>
<td>73.61</td>
</tr>
<tr>
<td>CAS</td>
<td>70.46</td>
<td>76.08</td>
</tr>
<tr>
<td>gCAS</td>
<td>73.08</td>
<td>75.78</td>
</tr>
</tbody>
</table>
### Precision, Recall, F1 score of turn-level act-slot pair

<table>
<thead>
<tr>
<th>method</th>
<th>movie</th>
<th>taxi</th>
<th>restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P$</td>
<td>$R$</td>
<td>$F_1$</td>
</tr>
<tr>
<td>classification</td>
<td>63.91</td>
<td>18.39</td>
<td>28.56</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>42.88</td>
<td>24.81</td>
<td>31.43</td>
</tr>
<tr>
<td>Copy Seq2Seq</td>
<td>41.90</td>
<td>23.12</td>
<td>29.80</td>
</tr>
<tr>
<td>CAS</td>
<td>43.12</td>
<td>31.60</td>
<td>36.47</td>
</tr>
<tr>
<td>gCAS</td>
<td>42.24</td>
<td>35.50</td>
<td><strong>38.58</strong></td>
</tr>
</tbody>
</table>
## Generated Examples

<table>
<thead>
<tr>
<th></th>
<th>example 1</th>
<th>example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>groundtruth</strong></td>
<td><code>request(date; starttime)</code></td>
<td><code>inform(restaurantname=; starttime =) multiple_choice(restaurantname)</code></td>
</tr>
<tr>
<td><strong>classification</strong></td>
<td><code>request+date</code></td>
<td><code>[]</code></td>
</tr>
<tr>
<td><strong>Seq2Seq</strong></td>
<td>‘request’ ‘(‘ date ‘;’ ‘starttime ‘)’</td>
<td>‘inform’ ‘(‘ restaurantname ‘=’ )’ ‘multiple_choice’ ‘=’ ‘restaurantname ‘)’</td>
</tr>
<tr>
<td><strong>Copy Seq2Seq</strong></td>
<td>‘request’ ‘(‘ date ‘=’ )’</td>
<td>‘inform’ ‘(‘ restaurantname ‘=’ ;’ ;’ ;’ ;’ ;’ =’ ;’ ;’ ‘starttime ‘=’ )’</td>
</tr>
<tr>
<td><strong>CAS</strong></td>
<td><code>request {}</code></td>
<td><code>inform {restaurantname}</code></td>
</tr>
</tbody>
</table>
| **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

Input Speech

User

Voice Output

Automatic Speech Recognition (ASR)

ASR Hypothesis

Spoken Language Understanding (SLU)

Domain / Intents / Slot Values

Dialog State Tracking (DST)

Dialog Policy

Belief State

Natural Language Generation (NLG)

System Response

Dialog Act

Text To Speech (TTS)

Text in

DB/KB
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.
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 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.
**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.

\[
P^c(w) = \begin{cases} 
  y^R_k, & \text{if } w = k^R, \\
  y^S_k, & \text{if } w = k^S, \\
  1, & \text{if } w \in I_t, \\
  0, & \text{otherwise}, 
\end{cases}
\]
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

\[ \mathcal{L} = \alpha^I \mathcal{L}^I + \alpha^R \mathcal{L}^R + \alpha^S \mathcal{L}^S + \alpha^A \mathcal{L}^A \]

Informable slot values
Requestable slots
Response slots
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 F₁ score of informable slot values and requestable slots.

For task completion evaluation, the Entity Match Rate (EMR) and Success F₁ score (SuccF₁) 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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inf P</th>
<th>Inf R</th>
<th>Inf F₁</th>
<th>Req P</th>
<th>Req R</th>
<th>Req F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSCP/RL†</td>
<td>0.970</td>
<td>0.971</td>
<td>0.971</td>
<td>0.983</td>
<td>0.935</td>
<td>0.959</td>
</tr>
<tr>
<td>TSCP†</td>
<td>0.970</td>
<td>0.971</td>
<td>0.971</td>
<td>0.983</td>
<td>0.938</td>
<td>0.960</td>
</tr>
<tr>
<td>FSDM/Res</td>
<td>0.979</td>
<td>0.984</td>
<td>0.978</td>
<td>0.994</td>
<td>0.947</td>
<td>0.967</td>
</tr>
<tr>
<td>FSDM</td>
<td><strong>0.983</strong></td>
<td><strong>0.986</strong></td>
<td><strong>0.984</strong></td>
<td><strong>0.997</strong></td>
<td><strong>0.952</strong></td>
<td><strong>0.974</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inf P</th>
<th>Inf R</th>
<th>Inf F₁</th>
<th>Req P</th>
<th>Req R</th>
<th>Req F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSCP/RL†</td>
<td><strong>0.936</strong></td>
<td>0.874</td>
<td>0.904</td>
<td>0.725</td>
<td>0.485</td>
<td>0.581</td>
</tr>
<tr>
<td>TSCP†</td>
<td>0.934</td>
<td>0.890</td>
<td>0.912</td>
<td>0.701</td>
<td>0.435</td>
<td>0.526</td>
</tr>
<tr>
<td>FSDM/Res</td>
<td>0.918</td>
<td>0.930</td>
<td>0.925</td>
<td>0.812</td>
<td>0.993</td>
<td>0.893</td>
</tr>
<tr>
<td>FSDM</td>
<td>0.92</td>
<td><strong>0.935</strong></td>
<td><strong>0.927</strong></td>
<td><strong>0.819</strong></td>
<td>1.000</td>
<td><strong>0.900</strong></td>
</tr>
</tbody>
</table>
## Dialogue-Level Task Completion Result

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CamRest</th>
<th></th>
<th>KVRET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>EMR</td>
<td>SuccF₁</td>
</tr>
<tr>
<td>Method</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NDM</td>
<td>0.212</td>
<td>0.904</td>
<td>0.832</td>
</tr>
<tr>
<td>LIDM</td>
<td>0.246</td>
<td>0.912</td>
<td>0.840</td>
</tr>
<tr>
<td>KVRN</td>
<td>0.134</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TSCP</td>
<td>0.253</td>
<td>0.927</td>
<td>0.854</td>
</tr>
<tr>
<td>TSCP/RL †</td>
<td>0.237</td>
<td>0.915</td>
<td>0.826</td>
</tr>
<tr>
<td>TSCP †</td>
<td>0.237</td>
<td>0.913</td>
<td>0.841</td>
</tr>
<tr>
<td>FSDM/St</td>
<td>0.245</td>
<td>-</td>
<td>0.847</td>
</tr>
<tr>
<td>FSDM/Res</td>
<td>0.251</td>
<td>0.924</td>
<td>0.855</td>
</tr>
<tr>
<td>FSDM</td>
<td><strong>0.258</strong></td>
<td><strong>0.935</strong></td>
<td><strong>0.862</strong></td>
</tr>
</tbody>
</table>
**Example of generated dialogue state and response**  
*(calendar scheduling domain)*

<table>
<thead>
<tr>
<th>user msg</th>
<th>what is the date and time of my next meeting and who will be attending it?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>belief state</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLD</strong></td>
<td>informable slot (event=meeting), requestable slot (date, time, party)</td>
</tr>
<tr>
<td><strong>TSCP</strong></td>
<td>‘meeting’ ‘(EOS_Z1)’ ‘date’ ‘;’ ‘party’</td>
</tr>
<tr>
<td><strong>FSDM</strong></td>
<td>event=meeting date=True time=True party = True</td>
</tr>
<tr>
<td><strong>agent response</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLD</strong></td>
<td>your next meeting is with party_SLOT on the date SLOT at time SLOT.</td>
</tr>
<tr>
<td><strong>TSCP</strong></td>
<td>your next meeting is at time SLOT on date SLOT at time SLOT.</td>
</tr>
<tr>
<td><strong>FSDM</strong></td>
<td>you have a meeting on date SLOT at time SLOT with party SLOT</td>
</tr>
</tbody>
</table>
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


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.


