

# Sequence-to-Sequence Learning for Task-oriented Dialogue with Dialogue State Representation

Haoyang Wen, Yijia Liu, Wanxiang Che\*, Libo Qin, Ting Liu  
Research Center for Social Computing and Information Retrieval,  
Harbin Institute of Technology, China  
{hywen, yjliu, car, lbqin, tliu}@ir.hit.edu.cn

## Abstract

Classic pipeline models for task-oriented dialogue system require explicit modeling the dialogue states and hand-crafted action spaces to query a domain-specific knowledge base. Conversely, sequence-to-sequence models learn to map dialogue history to the response in current turn without explicit knowledge base querying. In this work, we propose a novel framework that leverages the advantages of classic pipeline and sequence-to-sequence models. Our framework models a dialogue state as a fixed-size distributed representation and use this representation to query a knowledge base via an attention mechanism. Experiment on Stanford Multi-turn Multi-domain Task-oriented Dialogue Dataset shows that our framework significantly outperforms other sequence-to-sequence based baseline models on both automatic and human evaluation.

## Title and Abstract in Chinese

面向任务型对话中基于对话状态表示的序列到序列学习

面向任务型对话中，传统流水线模型要求对对话状态进行显式建模。这需要人工定义对领域相关的知识库进行检索的动作空间。相反地，序列到序列模型可以直接学习从对话历史到当前轮回复的一个映射，但其没有显式地进行知识库的检索。在本文中，我们提出了一个传统流水线与序列到序列二者优点的模型。我们的模型将对话历史建模为一组固定大小的分布式表示。基于这组表示，我们利用注意力机制对知识库进行检索。在斯坦福多轮多领域对话数据集上的实验证明，我们的模型在自动评价与人工评价上优于其他基于序列到序列的模型。

## 1 Introduction

Task-oriented dialogue system attracts more and more attention with the success of commercial systems such as Siri, Cortana and Echo. It helps users complete specific tasks with natural language. Figure 1 shows a typical example of a task-oriented dialogue, where an agent provides with location information for a user. The requirements for the agents to accomplish users' demands usually involve querying the knowledge base (KB), like acquiring address from location information KB in Figure 1.

Typical machine learning approaches model the problem as a partially observable Markov Decision Process (POMDP) (Williams and Young, 2007; Young et al., 2013), where a pipeline system is introduced. The pipeline system consists of four components: natural language understanding (NLU, Tur and De Mori, 2011), dialogue state tracking (Williams et al., 2013; Williams, 2012), dialogue policy learning (Young et al., 2010) and natural language generation (Wen et al., 2015). Taking the utterance in Figure 1 for example, NLU maps the utterance "Address to the gas station" into semantic slot "POI type". Dialogue state tracker keeps the probability of "gas station" close to 1 against other values of slot "POI type". Given a semantic frame as a dialogue state, which is the combination of distributions of these slots, dialogue policy learning generates the next pre-defined system action,

This work is licenced under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>

\* Email corresponding.

Address	Distance	POI type	POI	Traffic info
638 Amherst St	3 miles	grocery store	Sigona Farmers Market	car collision nearby
269 Alger Dr	1 miles	coffee or tea place	Cafe Venetia	car collision nearby
5672 barringer street	5 miles	certain address	5672 barringer street	no traffic
200 Alester Ave	2 miles	gas station	Valero	road block nearby
899 Ames Ct	5 miles	hospital	Stanford Childrens Health	moderate traffic
481 Amaranta Ave	1 miles	parking garage	Palo Alto Garage R	moderate traffic
145 Amherst St	1 miles	coffee or tea place	Teavana	road block nearby
409 Bollard St	5 miles	grocery store	Willows Market	no traffic

**Driver:** Address to the gas station.

**Car:** Valero is located at 200 Alester Ave.

**Driver:** OK , please give me directions via a route that avoids all heavy traffic.

**Car:** Since there is a road block nearby, I found another route for you and I sent it on your screen.

**Driver:** Awesome thank you.

Figure 1: An example of a task-oriented dialogue that incorporates a knowledge base. The knowledge base will be changed based on different dialogue environment setting. Agents need to generate response based on current knowledge base.

which usually involves querying the knowledge base. The action is then converted to its natural language expression using natural language generation. Both natural language understanding and dialogue state tracking require a large amount of domain specific annotation for training, which is expensive to obtain. Besides, the design of actions and the explicit forms of semantic frames require a lot of knowledge from human experts, which are domain-specific as well.

Neural generative models, typically Seq2Seq models, have achieved success on machine translation (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015). This success spurs the interests to apply Seq2Seq models into dialogue systems. Seq2Seq models map dialogue history directly into the response in current turn, while requires a minimum amount of hand-crafting. However, conventional Seq2Seq doesn't model the exterior data retrieval explicitly, which makes it hard for Seq2Seq to generate information stored in KB like meeting time and address, but this kind of retrieval is easy to achieve for classic pipeline. To tackle with the problem, Eric and Manning (2017) use an additional copy mechanism to retrieve entities that occurs in both KB and dialogue history. Eric et al. (2017) further introduced retrieval from key-value KB where the model uses key representations to retrieve the corresponding values. However, not all KBs are presented in key-value forms. Besides, an important component of classic pipeline, dialogue state tracker, is not properly modeled, making it difficult to precisely retrieve from KB.

In this paper, we propose a novel framework that takes the advantages from both classic pipeline models and Seq2Seq models. We introduce dialogue states into Seq2Seq learning, but in a implicit way. Distributions in classic state tracking are modeled as a group of representation vectors computed by an attention-based network (Britz et al., 2017), which can be considered as a dialogue state representation that aggregates information for each slot. And training this representation doesn't require annotation of dialogue state tracking. Our model queries the KB entries in an attention-based method as well, so that the querying is differentiable, without domain-specific pre-defined action spaces. Meanwhile we compute the representation for KB using entry-level attention and aggregate the representation with dialogue state representation to form a memory matrix of dialogue history and KB information. While decoding, we perform an attention over memory and an attention over input, incorporating copying mechanism (Gu et al., 2016) that allows model to copy words from KBs to enhance the capability of retrieving accurate entities.

We evaluate the proposed framework on Stanford Multi-turn, Multi-domain Dialogue Dataset (Eric et al., 2017), to test the effectiveness of our framework and flexibility to apply to different domains. We compare our model with other Seq2Seq models and discovered that our model has outperformed other

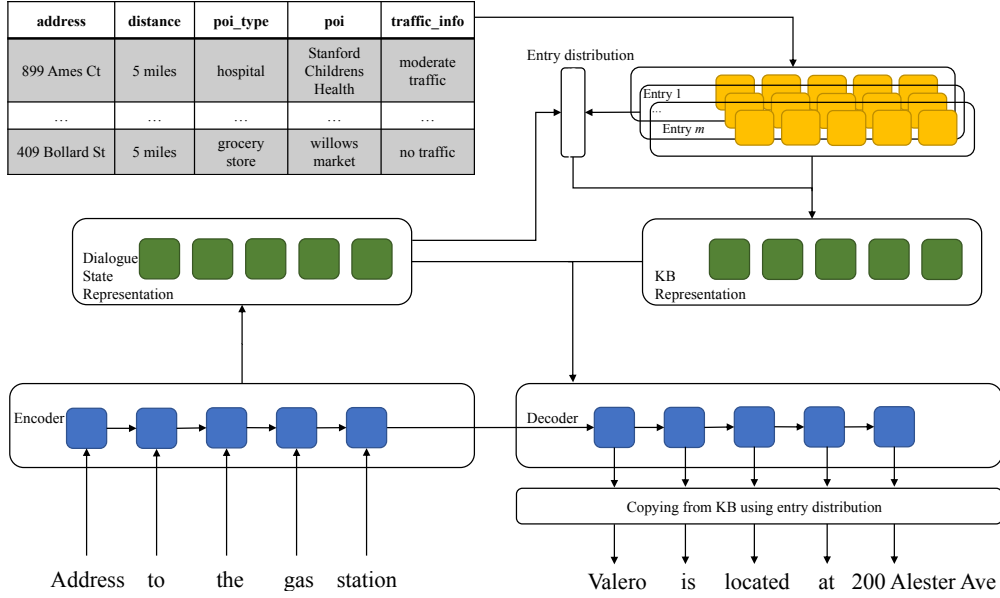


Figure 2: Proposed framework.

models on both automatic evaluation and human evaluation.

## 2 Proposed Framework

In this section, we describe a framework for task-oriented dialogue system. Our framework first encodes previous dialogue history, and computes dialogue state representation. Then our framework queries the table by attention and computes a matrix to represent information from previous history and KB. At last, the responses are generated using copying mechanism. The general architecture is demonstrated in Figure 2.

### 2.1 Encoder

Given a dialogue consisting of utterances from a user and an agent, our encoder encodes the whole dialogue history. We represent the dialogue history as a sequence of utterances. We encode the previous dialogue history as one single sequence consisting of each word in previous dialogue history and use  $(x_1, x_2, \dots, x_{n^{\text{IN}}})$  to denote the whole dialogue history word by word, where  $n^{\text{IN}}$  is the length of this sequence. Words are mapped to word embeddings and a long short-term memory network (LSTM) aggregates hidden representation over the sentence, denoted as  $\mathbf{H}^{\text{ENC}} = [h_1^{\text{ENC}}, \dots, h_{n^{\text{IN}}}^{\text{ENC}}]$  ( $\mathbf{H}^{\text{ENC}} \in \mathbb{R}^{d \times n^{\text{IN}}}$ ,  $d$  is the dimensionality of a hidden state).

### 2.2 Dialogue State Representation

The dialogue state tracking component of a dialogue system interprets the previous dialogue history to a belief state (Williams et al., 2013), which consists of a group of probability distributions over values for each slot. The dialogue state tracking is the core component of pipeline model. It helps with retrieving values from KB and generating accurate entities. To introduce dialogue state into Seq2Seq learning, in this framework, we model representation of belief state, motivated by Britz et al. (2017). We do not compute the explicit probability distribution for each slot. Instead, a group of distributed representations is computed. In this paper, we assume each turn of the dialogue is associated with  $m$  slots and its dialogue state representation is a matrix  $\mathbf{U}^{\text{IN}} = [\mathbf{u}_1^{\text{IN}}, \dots, \mathbf{u}_m^{\text{IN}}] \in \mathbb{R}^{d \times m}$  whose columns represent corresponding distribution. We further assume that  $m$  equals to the number of columns in our KB. Each state representation  $\mathbf{u}_k^{\text{IN}}$  is calculated by an attention over the whole representation of history:

$$\mathbf{u}_k^{\text{IN}} = \sum_{t=1}^{n^{\text{IN}}} a_k^{\text{IN}}(t) \mathbf{h}_t^{\text{ENC}},$$

where we assign each hidden state  $m$  different kinds of scores and perform a weighted average. The scores are computed by the following equations:

$$\begin{aligned} \text{SCOREIN}(\mathbf{w}_k^A, \mathbf{h}_t^{\text{ENC}}) &= \mathbf{w}_k^{A^T} \mathbf{h}_t^{\text{ENC}}, \\ a_k^{\text{IN}}(t) &= \frac{\exp(\text{SCOREIN}(\mathbf{w}_k^A, \mathbf{h}_t^{\text{ENC}}))}{\sum_{t'} \exp(\text{SCOREIN}(\mathbf{w}_k^A, \mathbf{h}_{t'}^{\text{ENC}}))}, \end{aligned}$$

$\mathbf{W}^A = [\mathbf{w}_1^A, \dots, \mathbf{w}_m^A]$  is a parameter matrix in  $\mathbb{R}^{d \times m}$ .

### 2.3 Soft KB Attention

**Table Encoder.** In our framework, we compile the process of querying a KB entry into an attention network. The first step is to encode the tables. Each KB cell is represented as the concatenation of the column name embedding  $\phi^{\text{EMB}}(f)$  and the cell value embedding  $\phi^{\text{EMB}}(c)$ . This representation is further fed into a tanh non-linearity and the final representation can be formalized as  $\mathbf{c} = \tanh(\mathbf{W}^C [\phi^{\text{EMB}}(c), \phi^{\text{EMB}}(f)])$ . The representation of the KB entry  $\mathbf{C}_k$  is denote as  $\mathbf{C}_k = [\mathbf{c}_{k,1}, \dots, \mathbf{c}_{k,m}]$ , consisting all its cell representations.

**Entry and KB Representation.** Conventionally, task-oriented dialogue systems interact with KB via carefully hand-crafted API calls, which are usually domain-specific and break the differentiability (Wen et al., 2017b). To make it differentiable, our framework applies a soft-attention over the KB entries and the attention value can be interpreted as the possibility that an entry will be used for decoding. We use the similarity between  $\mathbf{C}_k$  and  $\mathbf{U}^{\text{IN}}$  to represent attention score for the  $k^{\text{th}}$  entry. The similarity is computed by the following equation:

$$\text{sim}(\mathbf{C}_k, \mathbf{U}^{\text{IN}}) = \sum_{t=1}^m \mathbf{c}_{k,t} \cdot \mathbf{u}_t^{\text{IN}},$$

where we sum the dot product between vectors in  $\mathbf{C}_k$  and  $\mathbf{U}^{\text{IN}}$  respectively.

This similarity distribution is then converted into a probability distribution naturally using a softmax function. This probability distribution indicates the possibility to use entry  $e_k$  in the further response generation given previous dialogue history  $\mathbf{x}_{<i}$ :

$$p(e = e_k | \mathbf{x}_{<i}) = \frac{\exp(\text{sim}(\mathbf{C}_k, \mathbf{U}^{\text{IN}}))}{\sum_{k'} \exp(\text{sim}(\mathbf{C}_{k'}, \mathbf{U}^{\text{IN}}))}.$$

The information matrix from KB that is used for future generation can be computed as a weighted summation:

$$\mathbf{U}^{\text{KB}} = \sum_{t=1}^{|T|} p(e = e_t | \mathbf{x}_{<i}) \mathbf{C}_t,$$

where  $\mathbf{U}^{\text{KB}}$  is denoted as the information matrix,  $|T|$  is the number of entries in this KB.

Since the dimensionalities of all parameters are not related to the size of knowledge base. it allows changing the KB on-the-fly. Besides, there is no need to define specific operations, which is required for using API calls to extract information. Since we model the entries directly, it is appropriate to extract information from non-entity-centric knowledge bases as well.

Finally, we combine these two kinds of information by concatenating corresponding vectors and feeding them into a linear layer with an activation function. Formally, we denote  $\mathbf{U}^{\text{CAT}}$  as the concatenation of  $\mathbf{U}^{\text{IN}}$  and  $\mathbf{U}^{\text{KB}}$ . The two matrices are concatenated by concatenated corresponding vectors respectively. The process can be formulated as:  $\mathbf{U} = \tanh(\mathbf{W}^{\text{CAT}} \mathbf{U}^{\text{CAT}})$ . The matrix  $\mathbf{U}$  can be considered as a fix-sized memory representation over dialogue history and knowledge base.

## 2.4 Decoder

In this section, we will discuss the decoder that takes all previously calculated information to generate sentences. The vanilla Seq2Seq decoder generates a sequence of words recurrently based on the last hidden state of an encoder. We denote  $(\mathbf{h}_1^{\text{DEC}}, \dots, \mathbf{h}_{n_{\text{OUT}}}^{\text{DEC}})$  as the hidden states of the decoder and  $(y_1, \dots, y_{n_{\text{OUT}}})$  as the output sentence. We will consider two kinds of information, that are information of dialogue history and information from KB. The model aggregates information via attention over KB representation and history representation.

**Input Attention.** The conventional attention mechanism is introduced to extend the decoder, where each hidden state in the encoder is assigned a score based on the current hidden state  $\mathbf{h}_t^{\text{OUT}}$  at time step  $t$ , and then the context vector is computed by the weight summation (Luong et al., 2015). This process can be described by the following equations:

$$\begin{aligned} \mathbf{c}_t^{\text{IN}} &= \sum_{i=1}^{n^{\text{IN}}} a_t^{\text{OUT}}(i) \mathbf{h}_i^{\text{ENC}}, \\ \text{SCOREOUT}(\mathbf{h}_i^{\text{ENC}}, \mathbf{h}_t^{\text{DEC}}) &= \mathbf{v}^{\text{OUT}} \tanh(\mathbf{W}^{\text{OUT}} [\mathbf{h}_i^{\text{ENC}}, \mathbf{h}_t^{\text{DEC}}]), \\ a_t^{\text{OUT}}(i) &= \frac{\exp(\text{SCOREOUT}(\mathbf{h}_i^{\text{ENC}}, \mathbf{h}_t^{\text{DEC}}))}{\sum_{i'} \exp(\text{SCOREOUT}(\mathbf{h}_{i'}^{\text{ENC}}, \mathbf{h}_t^{\text{DEC}}))}. \end{aligned}$$

**Memory Attention.** Besides the context vector through input attention, we also use another context vector from the attention over the fix-size memory matrix  $\mathbf{U}$  and it's computed as:

$$\begin{aligned} \mathbf{c}_t^{\text{MEM}} &= \sum_{i=1}^m a_t^{\text{MEM}}(i) \mathbf{u}_i, \\ \text{SCOREMEM}(\mathbf{u}_i, \mathbf{h}_t^{\text{DEC}}) &= \mathbf{v}^{\text{MEM}} \tanh(\mathbf{W}^{\text{MEM}} [\mathbf{u}_i, \mathbf{h}_t^{\text{DEC}}]), \\ a_t^{\text{MEM}}(i) &= \frac{\exp(\text{SCOREMEM}(\mathbf{u}_i, \mathbf{h}_t^{\text{DEC}}))}{\sum_{i'} \exp(\text{SCOREMEM}(\mathbf{u}_{i'}, \mathbf{h}_t^{\text{DEC}}))}. \end{aligned}$$

In practice, we use an additional context vector from encoder to calculate the dialogue state representation, which is different from the one that is used for the start of decoding.

The two kinds of context vectors and the current hidden state are used for decoding. We introduce a variant of copying mechanism (Gu et al., 2016) in order to retrieve entities from KB directly. We first compute a probability distribution over output vocabulary  $\mathcal{V}$  and slot types  $\mathcal{V}^{\text{SLOT}}$ , given previous dialogue history  $\mathbf{x}_{<i}$  and previous generated words  $\mathbf{y}_{<t}$ , which can be described as:  $p(\tilde{y}_t | \mathbf{x}_{<i}, \mathbf{y}_{<t}) = \text{softmax}(W^{\text{O}} [\mathbf{h}_t^{\text{DEC}}, \mathbf{c}_t^{\text{IN}}, \mathbf{c}_t^{\text{MEM}}])$ . The probability of generating a slot type represents the sum of probability of generating a slot value for this slot from KB. Since we have calculated the probability of using an entry in section 2.3, the probability of generating a word can be described as:

$$\begin{aligned} p(y_t = y | \tilde{y}_t, \mathbf{x}_{<i}, \mathbf{y}_{<t}) &= p(\tilde{y}_t = y | \mathbf{x}_{<i}, \mathbf{y}_{<t}) \\ &+ \sum_{y^{\text{S}} \in \mathcal{V}^{\text{SLOT}}} p(\tilde{y}_t = y^{\text{S}} | \mathbf{x}_{<i}, \mathbf{y}_{<t}) \sum_{k=1}^{|T|} \mathbb{1}\{e_k(y^{\text{S}}) = y\} p(e = e_k | \mathbf{x}_{<i}), \end{aligned}$$

where  $e_k(y^{\text{S}})$  is the cell that is in slot (i.e. column)  $y^{\text{S}}$ . Note that the summation of probability over  $\mathcal{V}$  is exactly 1 after the model copies entities from KB.

## 2.5 Training

Conventionally, we use negative log likelihood (NLL) for training to train a Seq2Seq model. Since there is no supervision for soft KB attention, and it is easy to get over-fitting when we only use NLL, we

	<b>Weather</b>	<b>Navigation</b>
<b>Slot Types</b>	location, date, highest temperature, lowest temperature and weather attribute	POI name, traffic info, POI type, address, distance

Table 1: Slot types for different domains.

apply policy gradient to improve the performance of soft KB attention as well. We consider the KB and fix-size memory representation from input as the environment in a reinforcement learning setting. There is only one action, which is defined as choosing a single entry from KB to help with generating response. Heuristically, the more entities in an entry appear in dialogue context, the higher possibility that this entry is used for generating response. Therefore, we consider the number of entities from an entry  $e$  that appear in previous dialogue history or current gold response as reward  $R(e)$ , and apply REINFORCE with baseline (Williams, 1992):  $J_{RL} = -\mathbb{E}_{p(e|x_{<i>i})} [R(e) - b]$ , where  $b$  is a hyperparameter denoting the baseline reward. The joint loss is the combination of the NLL and loss from reinforcement learning, which is:

$$J = J_{NLL} + \lambda J_{RL}$$

where  $\lambda$  is a hyperparameter balancing the two factors. In practice, we first use  $J_{RL}$  only to train the soft KB attention and its encoder, without training for response generation, which will accelerate the convergence of Seq2Seq learning. We apply data augmentation to the original dataset as well, where we add delexicalised form responses into training data in order to force our model to generate slot types first and then retrieve entity from KB using copying mechanism. The delexicalised responses are generated by simple matching and replacing.

### 3 Experiment

In this section, we first introduce the details of experiment setting. Then we provide results and analyses of automatic evaluation and human evaluation in order to compare with other baseline models. Besides, we present ablation test to evaluate and analyze the function of different components in our framework. Finally, we provide visualization of dialogue state representation and case study.

#### 3.1 Experiment Setting

We choose two KB-rich domains from Stanford Multi-turn Multi-domain Task-oriented Dialogue Dataset Eric et al. (2017), which are weather information retrieval and point-of-interest (POI) navigation (navigation). We first change the form of KB in weather information retrieval domain (weather). Eric et al. (2017) integrates the highest temperature, lowest temperature and weather information into a single weekday column due to their incapability of utilizing non-entity-centric KB. In this paper, we separate these information into three different column, and the slots of these two domains are provided in Table 3.

Our framework is trained separately in these two domains, using the same train/validation/test split sets as Eric et al. (2017). We do not map the entities in dialogue into its canonical form as what Eric et al. (2017) have done, since our framework extract entities directly from KB. And we evaluate our framework on exact entities as well.

Our framework is trained using the Adam optimizer (Kingma and Ba, 2014). The learning rate is  $10^{-3}$ ,  $\lambda$  for balancing two loss functions is  $10^{-1}$ . We applied dropout (Srivastava et al., 2014) to the input and the output of LSTM, with a dropout rate at 0.75. We add the weight decay on the model. The coefficient of weight decay is  $5 \times 10^{-6}$ . The embedding size and all hidden size are 200. The number of epochs for pretraining soft KB attention is 30 for ooth navigation and weather. Baseline  $b$  is 1.5 for both navigation and weather.

##### 3.1.1 Baseline Models

We compare our model with two additional baselines beyond the Seq2Seq with attention, which includes:

- **Copy-augmented Sequence-to-Sequence Network.** This model is adapted from (Eric and Manning, 2017). It utilizes entities that appear in both previous dialogue history and KB. A hard copy mechanism for these entities is applied in this model.
- **Key-value Retrieval Network.** This model is adapted from Eric et al. (2017). It utilizes key-value forms to represent KBs. Key representations are used for an attention-based value retrieval. In weather information retrieval domain, although we have changed the KB into a non-entity-centric form, we still designate “location” slot as subject slot and we allow a key representation to retrieval multiple values. We convert inputs into canonical forms, while the outputs remain the same in order to compare with our model.

## 3.2 Automatic Evaluation

In this section, we provide two different automatic evaluations to compare with other baseline models. The results and analyses are provide in the following sections.

### 3.2.1 Evaluation Metrics

Entity F1 and BLEU score are used to evaluate our model. The entity F1 scores in both micro-average and macro-average manners are used to measure the difference between entities in the system and gold responses. Besides, we use BLEU to evaluate the quality of responses. Sharma et al. (2017) showed that BLEU has a comparatively strong correlation with human evaluation on task-oriented dialogue dataset. The BLEU score is computed from results with highest micro F1. To evaluate macro F1, we delete instances that neither gold nor generated response contains an entity.

### 3.2.2 Results and Analyses

Experiment results are illustrated in Table 2. The results show that our model outperforms other models in most of automatic evaluation metrics. In the navigation domain, compared to KV Net, we achieve 5.0 improvement on BLEU score, 37.1 improvement on Macro F1 and 27.4 improvement on Micro F1. Compared to Copy Net, we achieve 5.0 improvement on BLEU score, 41.2 improvement Macro F1 and 27.4 improvement on Micro F1. The results in navigation show our model’s capability to generate more natural and accurate response than the Seq2Seq baseline models. In the weather domain, our model generates more accurate responses than our baseline models as well. The BLEU score is a little bit lower than Copy Net and Seq2Seq with attention. This is because the forms of responses are relatively limited in weather domain. Besides, the entities in inputs are highly probable to be mentioned in responses, such as “location”. These two reasons indicate that the simpler models can capture this pattern more smoothly. The results that Seq2Seq with Attention performs better than Copy Net and KV Net also confirm this.

We also find that the KV Net’s results are lower than that reported by Eric et al. (2017). We address this to the differences in the preprocessing, model training and evaluation metrics. In spite of the difference of evaluation metrics that we evaluate on exact entities rather than their canonical forms, the Micro F1 score of our model still outperforms what Eric et al. (2017) reported, which is 41.3 in navigation domain and which is evaluated on canonical forms. Our changes of the weather domain into non-entity-centric also influence its performance. This differences in results also indicate the robustness of our model when facing non-entity-centric KBs.

### 3.2.3 Ablation

In this section, we perform several ablation experiments to evaluate different components in our framework on the navigation domain. Results are shown in Table 3. The results demonstrate effectiveness of components of our model to the final performance.

Copying mechanism enables our framework to retrieve entities directly from KBs. Without copying mechanism, such retrieval is infeasible and our framework cannot produce values in KBs. The results show that it introduces more variability to the generation process if we do not use copying mechanism.

The reinforcement learning loss helps our framework to use correct KB entries so that improve the performance of generation. Without this reinforcement learning loss, the item selection process is only

Model	Navigation			Weather		
	BLEU	Macro F1	Micro F1	BLEU	Macro F1	Micro F1
Seq2Seq with Attention	8.3	15.6	17.5	<b>19.6</b>	56.0	53.5
Copy Net	8.7	20.8	23.7	17.5	52.4	53.1
KV Net	8.7	24.9	29.5	12.4	37.7	39.4
our model	<b>13.7</b>	<b>62.0</b>	<b>56.9</b>	14.9	<b>58.5</b>	<b>56.3</b>

Table 2: Automatic evaluation on test data. Best results are shown in bold. Generally, our framework outperforms other models in most automatic evaluation metrics.

Model	BLEU	Macro F1	Micro F1
our model	<b>13.7</b>	<b>62.0</b>	<b>56.9</b>
-copying	9.6	35.2	41.3
-RL	9.3	38.2	46.0

Table 3: Ablation experiment on navigation domain. -copy refers to a framework without copying. -RL refers to a framework without RL loss.

Model	Correct	Fluent	Humanlike
Copy Net	3.52	4.47	4.17
KV Net	3.61	4.50	4.20
our model	<b>4.21</b>	<b>4.65</b>	<b>4.38</b>
agreement	41.0	55.0	43.0

Table 4: Human evaluation of responses based on random selected previous dialogue history in test dataset. The agreement scores indicate the percentage of responses to which all three human experts give exactly the same scores.

supervised by log likelihood loss. We address the drop in performance to that our model overfits to the training data.

### 3.3 Human Evaluation

In this section, we provide human evaluation on our framework and other baseline models. We randomly generated 200 responses. These response are based on distinct dialogue history in navigation test data. We hire three different human experts to evaluate the quality of responses. Three dimensions are involved, which are correctness, fluency, and humanlikeness. Three human experts judged each dimension on a scale from 1 to 5. And each judgment indicates a relative score compared to standard response from test data. The results are illustrated in Table 4. The results show that our framework outperforms other baseline models on all metrics. The most significant improvement is from correctness, indicating that our model generates more accurate information that the users want to know.

### 3.4 Visualization and Case Study

We provide a visualization example to demonstrate the effectiveness of dialogue state representation. The visualization illustrates attention scores over the sentence for each slot. The blue-level of a cell indicates the attention score it represents. From this visualization, we can discover that our dialogue state representation matches slots with correct entities in sentence. For example, “pizza\_restaurant” matches “poi” and “poi\_type” correctly, “4\_miles” matches “address” correctly. The visualization indicates the capability of our framework to track accurate information and integrate them into a fix-size matrix representation.

Finally, we provide a case study that consists of two conversations which are generated between our framework and a human and between Seq2Seq with Attention with human respectively. In this case, we find that our framework is able to generate correct information such as address and point-of-interest.



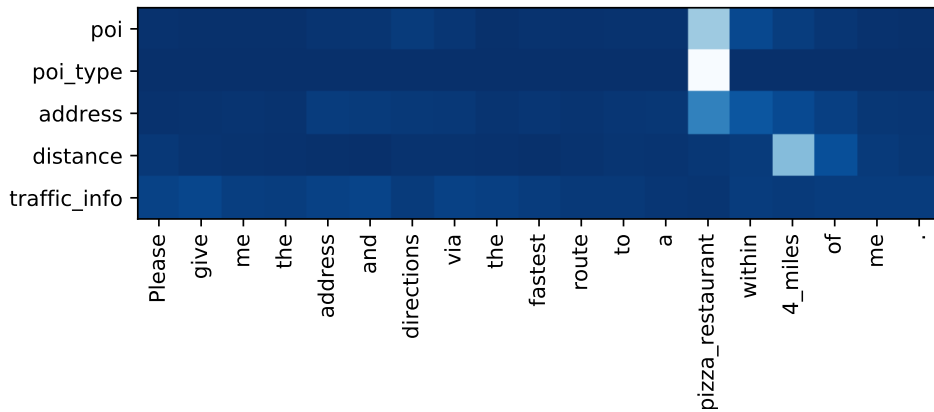


Figure 3: Visualization of dialogue state representation.

Address	Distance	POI type	POI	Traffic info
783 Arcadia Pl	5 miles	gas station	Chevron	moderate traffic
383 University Ave	5 miles	shopping center	Town and Country	no traffic
864 Almanor Ln	5 miles	friends house	jacks house	no traffic
5671 barringer street	6 miles	home	home	heavy traffic
657 Ames Ave	4 miles	rest stop	The Clement Hotel	no traffic
638 Amherst St	1 miles	grocery store	Sigona Farmers Market	heavy traffic
830 Almanor Ln	6 miles	chinese restaurant	tai pan	no traffic

our model		Seq2Seq with Attention	
<b>Driver:</b>	what gas stations are here ?	<b>Driver:</b>	what gas stations are here ?
<b>Car:</b>	there is a chevron 5 miles away .	<b>Car:</b>	there is a chevron 4 miles away .
<b>Driver:</b>	that 's good ! please show me the address and pick the quickest route to get there !	<b>Driver:</b>	that 's good ! please show me the address and pick the quickest route to get there !
<b>Car:</b>	the address is 783 arcadia pl , i sent on your screen the best route to reach there .	<b>Car:</b>	the address to safeway is 452 arcadia pl .
<b>Driver:</b>	thank you !	<b>Driver:</b>	thank you !
<b>Car:</b>	you 're welcome !	<b>Car:</b>	you 're welcome !

Figure 4: Case study. The dialogue in the left is generated from our framework, and that in the right is generated from Seq2Seq with Attention baseline.

Conversely, Seq2Seq with Attention is generated in a more random way. The comparison between these two dialogues illustrates the capability of our model to retrieve accurate entities and while in the same time generate natural response.

#### 4 Related Work

The recent successes on neural networks spread across many natural language processing tasks, which also stimulate research on task-oriented dialogue system. The powerful distributed representation ability of neural networks makes task-oriented dialogue system end-to-end possible. Recently, Wen et al. (2017b) built a system that connects classic pipeline modules by a policy network training with a user simulator. Wen et al. (2017a) further introduced latent intention into end-to-end learning. However, their modules like belief tracker still needs to be trained separately before end-to-end training. In contrast to their work, our framework trained the state tracker jointly with the end-to-end dialogue training. Liu and

Lane (2017) built a turn-level LSTM to model the dialogue state and generate probability distribution for each slot. Bordes and Weston (2017) built a system by applying memory network to store the previous dialogue history. But the responses are retrieved from templates, which is significantly different from our neural generative responses. Another type of work tried to build an end-to-end system as a task completion dialogue system (Li et al., 2016; Li et al., 2017; Peng et al., 2017). These modeled are trained through an end-to-end fashion via reinforcement learning algorithm using the reward from user simulators. However, their dialogue responses are not generated from the dialogue history directly but from a set of pre-defined action and explicit semantic frames.

Our soft KB attention can be considered as a process to retrieve entries, which has been explored in many QA and dialogue work. One line of this research includes creating well-defined API calls to query the KB (Williams et al., 2017; Wen et al., 2017a). And another line of research tried to directly retrieve entities from knowledge base. Yin et al. (2016b) has built a system to encode all table cells and assign a score vector to each row. Our framework resembles the second line of research, but can generate multiple entities to form natural language responses. He et al. (2017) has built two symmetric dialogue agents with private knowledge, and has applied knowledge graph reasoning into Seq2Seq learning, which is distantly related with our framework. In the sense of the KB forms, Yin et al. (2016a) retrieved entities based on (*subject, relation, object*) triples. While Dhingra et al. (2017) applied a soft-KB lookup on an entity-centric knowledge base to compute the probability of that the user knows the values of slots, and has tried to model the posterior distributions over all slots. However, our framework doesn't require entity-centric knowledge base.

## 5 Conclusion

In this paper, we proposed a framework that leverages dialogue state representation, which is tracked by an attention-based methods. Our framework performed an entry-level soft lookup over the knowledge base, and applied copying mechanism to retrieve entities from knowledge base while decoding. This framework was trained in an end-to-end fashion with only the dialogue history, and get rid of other annotation. Experiments showed that our model outperformed other Seq2Seq models on both automatic and human evaluation. The visualization and case study demonstrated the effectiveness of dialogue state representation and entity retrieval.

## 6 Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61632011 and 61772153.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of International Conference on Learning Representations*.
- Denny Britz, Melody Guan, and Minh-Thang Luong. 2017. Efficient attention using a fixed-size memory representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 392–400.
- Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 484–495.
- Mihail Eric and Christopher Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: (Volume 2, Short Papers)*, volume 2, pages 468–473.

- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGDial Meeting on Discourse and Dialogue*, pages 37–49.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640.
- He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1766–1776.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.
- Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 733–743.
- Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *Interspeech 2017*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.
- Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240.
- Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.
- Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 275.
- Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017a. Latent intention dialogue models. In *International Conference on Machine Learning*, pages 3732–3741.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, 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 *EACL*, pages 438–449, Valencia, Spain, April. Association for Computational Linguistics.
- Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.
- Jason D Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the 14th Annual SIGDial Meeting on Discourse and Dialogue*, pages 404–413.
- Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 665–677.

- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.
- Jason D Williams. 2012. A belief tracking challenge task for spoken dialog systems. In *NAACL-HLT Workshop on future directions and needs in the spoken dialog community: tools and data*, pages 23–24. Association for Computational Linguistics.
- Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016a. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 2972–2978. AAAI Press.
- Pengcheng Yin, Zhengdong Lu, Hang Li, and Ben Kao. 2016b. Neural enquirer: learning to query tables in natural language. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. AAAI Press.
- Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.