INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Task-Oriented Dialog Generation with Enhanced Entity Representation

Zhenhao He[⋆], *Jiachun Wang*[⋆], *Jian Chen*[†]

School of Software Engineering, South China University of Technology, China
{sezhenhao.he,sewangjiachun}@mail.scut.edu.cn, ellachen@scut.edu.cn

Abstract

Recent advances in neural sequence-to-sequence models have led to promising results for end-to-end task-oriented dialog generation. Such frameworks enable a decoder to retrieve knowledge from the dialog history and the knowledge base during generation. However, these models usually rely on learned word embeddings as entity representation, which is difficult to deal with the rare and unknown entities. In this work, we propose a novel enhanced entity representation (EER) to simultaneously obtain context-sensitive and structure-aware entity representation. Our proposed method enables the decoder to facilitate both the ability to fetch the relevant knowledge and the effectiveness of incorporating grounding knowledge into the dialog generation. Experimental results on two publicly available dialog datasets show that our model outperforms the state-of-the-art data-driven task-oriented dialog models. Moreover, we conduct an Out-of-Vocabulary (OOV) test to demonstrate the superiority of EER in handling common OOV problem.

Index Terms: dialog systems, task-oriented dialog systems, entity representation, natural language generation

1. Introduction

Task-oriented dialog systems, which aim to help users to accomplish specific tasks via natural language, have become an increasingly important research area. With the success of the sequence-to-sequence (seq2seq) architecture in text generation [1, 2, 3, 4, 5], many works have attempted to model task-oriented dialog systems as the seq2seq generation of response from the dialog history and the external knowledge bases (KB) [6, 7, 8, 9, 10]. This kind of fully data-driven end-to-end modeling scheme eliminates the requirement of hand-crafted slot filling labels, which is a promising way to build domain-agnostic dialog systems.

Different from general seq2seq generation, knowledge-grounded dialog generation requires to understand the user intent, query external KB, and generate system responses with grounding knowledge. Moreover, the ability to fetch the right knowledge from KB is essential in task-oriented dialog systems, because the responses are guided not only by the dialog history but also by the query results [8]. One straightforward method of existing works is to learn word embeddings for each entity in the KB, and then encode the dialog history as query vector to obtain the most relevant knowledge [7, 11, 12]. In addition, another previously proposed method models KB with memory network [13], which encodes each knowledge as memory elements and uses multi-hop attention mechanism to select the appropriate knowledge [8, 9, 10].

However, although the above mentioned approaches are successful in incorporating knowledge into the dialog generation, they still suffer from heavily relying on learned word embeddings as entity representation. For one thing, some of the named entities in the KB or the dialog history occur less frequently in the training set (e.g., a particular organization name) and thus are difficult to learn a good word embedding, resulting in poor performance. For another, when the unseen entities are encountered at test time (also known as OOV), it is prone to lose some important information by mapping them to a special token. Moreover, with large KB in real-world scenarios, learning word embeddings for each entity will cause an explosion in the vocabulary size and the number of parameters. Regarding these issues, learning a more flexible and informative entity representation is very important.

To alleviate these limitations, we propose a novel enhanced entity representation (EER) for task-oriented dialog generation, which can deal with the rare or unseen entity problem. Our model separately treats entities from two sources, i.e., the dialog history and KB, and encodes these entities based on their own context or attribute information. Specifically, the entities can learn context-sensitive representation from the dialog context since the surrounding words of each entity can provide some meaningful and distinguishable information. On the other hand, the structural and relational information in the KB are able to endow each entity from the KB with structure-aware entity representation. Through the above EER, we can reduce the dependence of the learned entity embeddings and hence generalize to the rare and unseen entities. Moreover, we further propose a switching network to softly control the weight between context-sensitive and structure-aware entity representation, which better integrates knowledge into the dialog generation. Our experiments on two publicly available datasets (InCar Assistant dataset [7] and CamRest dataset [12]) show an improvement in both entity F1 scores, and BLEU scores as compared to existing state-of-the-art architectures. Our code is available on GitHub[1].

2. Task Definition

Before describing our proposed method, we formally define the dialog generation task. Given a dialog between a user (U) and a system (S), we represent the t-turned dialog utterances as $\{(U_1, S_1), ..., (U_t, S_t)\}$. Along with these utterances, each dialog is accompanied by a relational-database-like KB B, which consists of $|\mathcal{R}|$ rows and $|\mathcal{C}|$ columns. In this sense, entities may come from two sources, namely the dialog history that contains l entities $\{e_i^c\}_{i=1}^l$ and the KB with $e_{i,j}^k$ the entity in the i^{th} row and j^{th} column. At the t^{th} turn of the dialog, we take the dialog history $(U_1, S_1, ..., S_{t-1}, U_t)$ and the associated KB B as input, and our goal is to generate a proper system response S_t.

3. Our Proposed Method

Figure 1 illustrates the architecture of our proposed model, which consists of a context encoder, an entity encoder and a

[⋆] Both authors contributed equally.
[†] Corresponding Author.

[1] https://github.com/scoyer/EER

http://dx.doi.org/10.21437/Interspeech.2020-1037

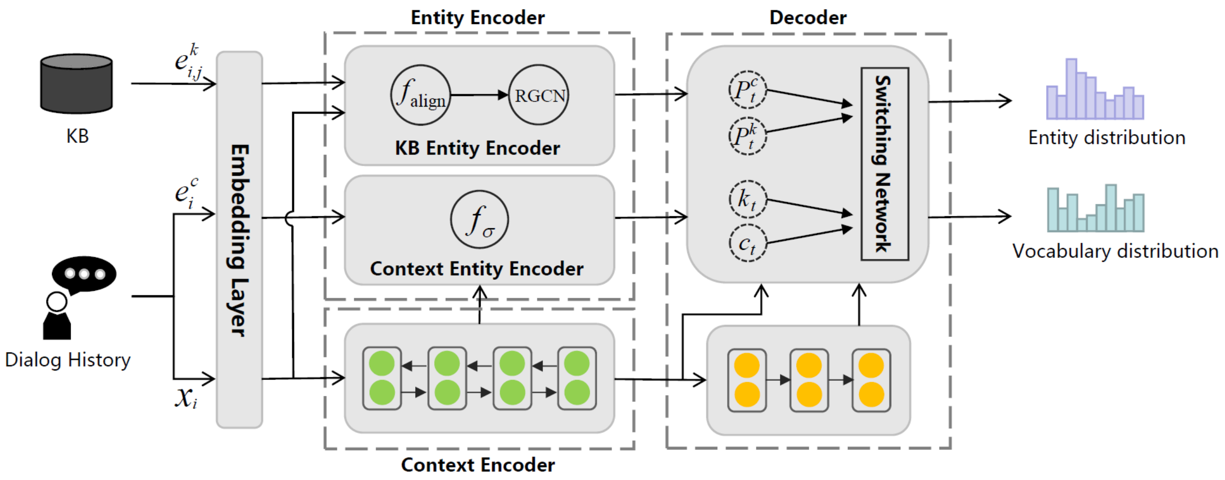

Figure 1: *Architecture of the proposed model.*

decoder. The context encoder encodes the dialog history into a sequence of hidden vectors, while the entity encoder produces context-sensitive representation for entities from dialog context and structure-aware representation for entities from KB. Finally, the decoder generate entity distribution and vocabulary distribution conditioned on the output of context and entity encoders.

3.1. Context Encoder

The context encoder concatenates all the m words in the dialog history as a sequence of tokens $\boldsymbol{x} = (x_1, ..., x_m)$, and then employs a bi-directional gated recurrent unit (GRU) [14] to transform the word sequence into a sequence of hidden vectors. Specifically, the forward GRU reads input sequence $\boldsymbol{x}$ in left-to-right direction while the backward GRU reads $\boldsymbol{x}$ in the reversed direction:

$$\overrightarrow{\boldsymbol{h}}_i = \overrightarrow{\text{GRU}}(\boldsymbol{E}(x_i), \overrightarrow{\boldsymbol{h}}_{i-1}), \tag{1}$$

$$\overleftarrow{\boldsymbol{h}}_i = \overleftarrow{\text{GRU}}(\boldsymbol{E}(x_i), \overleftarrow{\boldsymbol{h}}_{i+1}), \tag{2}$$

where $\boldsymbol{E}(x_i)$ is the word embedding of the token x_i. We obtain a hidden representation for each word x_i by concatenating the forward hidden state $\overrightarrow{\boldsymbol{h}}_i$ and the backward one $\overleftarrow{\boldsymbol{h}}_i$, i.e., $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i||\overleftarrow{\boldsymbol{h}}_i]$. Here, $||$ denotes the concatenation. Thus, each hidden vector $\boldsymbol{h}_i$ contains the information about the i^{th} word with respect to all the surrounding words in both directions.

3.2. Entity Encoder

Entity encoder is designed to obtain entity representation from two sources, namely the dialog history and KB. For entities in the dialog history, we propose a simple and effective way to produce context-sensitive representation. For entities in the KB, they are first passed to an alignment function and then learn structure-aware representation from the relational structure.

Context Entity Encoder. We first detect all the entities in the dialog history. Let $(e_1^c, ..., e_l^c)$ be the entities in the dialog history, and $(p_1, ..., p_l)$ be their corresponding positions in the input sequence $\boldsymbol{x}$. For each entity e_i^c, we first concatenate the entity embedding and its hidden state calculated by context encoder, and then pass it to a nonlinear mapping function:

$$\boldsymbol{h}_i^c = f_\sigma([\boldsymbol{E}(x_{p_i})||\boldsymbol{h}_{p_i}]), \tag{3}$$

where $f_\sigma(\cdot)$ is a single-layer feed-forward neural network with ReLU [15] nonlinearity. Through combining the learned word embedding and the contextual hidden states, entities with the same value have different context-sensitive representation. For the case of unseen entities, such entity representation is meaningful since it captures the contextual information.

KB Entity Encoder. We regard every value of the KB as an entity. Similar to the prior work [16], we introduce an aligned context embedding to add soft alignments between entity and similar words in the dialog history. Specifically, we define an alignment function $f_a(e_{i,j}^k) = \sum_t a_{i,j}^t \boldsymbol{E}(x_t)$, where the attention score $a_{i,j}^t$ captures the similarity between entity $e_{i,j}^k$ and word x_t. And $a_{i,j}^t$ is computed by the dot products between nonlinear mappings of word embeddings:

$$a_{i,j}^t = \frac{\exp(f_\sigma(\boldsymbol{E}(e_{i,j}^k)) \cdot f_\sigma(\boldsymbol{E}(x_t)))}{\sum_{t'} \exp(f_\sigma(\boldsymbol{E}(e_{i,j}^k)) \cdot f_\sigma(\boldsymbol{E}(x_{t'})))}. \tag{4}$$

Then we concatenate the entity embedding and its corresponding aligned context embedding, and pass it to a nonlinear mapping function:

$$\boldsymbol{h}_{i,j}^a = f_\sigma([\boldsymbol{E}(e_{i,j}^k)||f_a(e_{i,j}^k)]). \tag{5}$$

However, simply using the above alignment function to encode each entity ignores the rich structure inherently in the KB. It's important to exploit the underlying relational structure to obtain the structure-aware entity representation which improves the performance of modeling KB. To this end, we regard the entity to be encoded as a node in the graph and other entities in the same row as neighboring nodes, and apply Relational Graph Convolutional Networks (RGCNs) [17] to encode each entity:

$$\boldsymbol{h}_{i,j}^k = \sigma\left(\boldsymbol{W}_0 \boldsymbol{h}_{i,j}^a + \frac{1}{|N_{i,j}|} \sum_{j',j' \neq j} \boldsymbol{W}_{j'} \boldsymbol{h}_{i,j'}^a\right), \tag{6}$$

where $\{\boldsymbol{W}_j\}_{j=0}^{|\mathcal{C}|}$ are trainable parameters, $|N_{i,j}|$ is a normalization constant that represents the number of entities except for $e_{i,j}^k$ in the i^{th} row, and $\sigma(\cdot)$ is an element-wise activation function (e.g., ReLU). For each entity, it not only performs self-transformation with current hidden representation, but also receives and aggregates the information from other entities in the same row using relation-specific transformation. Thus, even if the entities with the same value, they can capture more diverse representation according to the relational structure.

3.3. Decoder

Given the output of context and entity encoders, the decoder uses a GRU to maintain a hidden sequence $\{\boldsymbol{s}_t\}_{t=1}^{n}$. Let $\boldsymbol{y} = (w_1, ..., w_n)$ be the referenced response, then $\boldsymbol{s}_t$ is defined by:

$$\boldsymbol{s}_t = \text{GRU}(\boldsymbol{E}(w_{t-1}), \boldsymbol{s}_{t-1}), \tag{7}$$

where the initial hidden state $\boldsymbol{s}_0$ is $f_\sigma(\boldsymbol{h}_m)$. Based on $\{\boldsymbol{s}_t\}_{t=1}^{n}$, the decoder generates two distributions, namely the entity distribution about one over the entity set which contains all the entities appearing in the dialog history and KB, denoted as P^{entity}, and the vocabulary distribution about one over all the generic words in the vocabulary, denoted as P^{vocab}.

Entity distribution consists of two parts: the context entity distribution regarding the entities from the dialog history, denoted as P^c, and the KB entity distribution about entities from the KB, denoted as P^k. At each time step, a match function is used to measure the similarity between hidden state of decoder and entity representation. Formally, for entities from the dialog history, we have:

$$P_t^c(e) = \sum_{i:e=e_i^c} \frac{\exp(\boldsymbol{s}_t^\top \boldsymbol{W}_c \boldsymbol{h}_i^c)}{\sum_{i'} \exp(\boldsymbol{s}_t^\top \boldsymbol{W}_c \boldsymbol{h}_{i'}^c)}, \tag{8}$$

where e represents entity in entity set and $\boldsymbol{W}_c$ is a similarity matrix to be learned. For entities from the KB, we employ hierarchical attention mechanism to calculate the probability for each entity from the KB. To detail, we first apply average pooling over entities in the same row to compute the row-level matching scores, and then calculate the entity-level matching scores:

$$p_i^r = \frac{\exp(\boldsymbol{s}_t^\top \boldsymbol{W}_k \bar{\boldsymbol{h}}_i^k)}{\sum_{i'} \exp(\boldsymbol{s}_t^\top \boldsymbol{W}_k \bar{\boldsymbol{h}}_{i'}^k)}, \tag{9}$$

$$p_{i,j}^e = \frac{\exp(\boldsymbol{s}_t^\top \boldsymbol{W}_k \boldsymbol{h}_{i,j}^k)}{\sum_{j'} \exp(\boldsymbol{s}_t^\top \boldsymbol{W}_k \boldsymbol{h}_{i,j'}^k)}, \tag{10}$$

where $\boldsymbol{W}_k$ is a similarity matrix to be learned and $\bar{\boldsymbol{h}}_i^k$ is the average pooling of $\{\boldsymbol{h}_{i,j}^k\}_{j=1}^{|\mathcal{C}|}$. Thus, we have:

$$P_t^k(e) = \sum_{i,j:e=e_{i,j}^k} p_i^r \cdot p_{i,j}^e. \tag{11}$$

Switching Network. Instead of fusing context entity distribution and KB entity distribution by simple element-wise addition, we propose a switching network in analogy to [18]. The switching network is a feed-forward neural network with sigmoid output function that outputs a scalar probability of switching between context entity distribution and KB entity distribution. Let g_t be the output probability, then we have:

$$g_t = \text{sigmoid}(\boldsymbol{W}_g \boldsymbol{s}_t + \boldsymbol{b}_g), \tag{12}$$

$$P_t^{\text{entity}} = g_t P_t^c + (1 - g_t) P_t^k, \tag{13}$$

where $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are trainable parameters.

The decoder predicts a word in the vocabulary by attending over the output of context encoder and entity encoder, since attention allows the decoder to dynamically decide the importance of context representation and entity representation. Formally,

$$\boldsymbol{c}_t = \text{attn}(\boldsymbol{s}_t, \{\boldsymbol{h}_i\}_{i=1}^{m}), \tag{14}$$

$$\boldsymbol{k}_t = g_t \sum_i p_i^c \boldsymbol{h}_i^c + (1 - g_t) \sum_i p_i^r \bar{\boldsymbol{h}}_i^k, \tag{15}$$

$$P_t^{\text{vocab}} = \boldsymbol{W}_v [\boldsymbol{s}_t || \boldsymbol{c}_t || \boldsymbol{k}_t], \tag{16}$$

where attn(*query*,*memory*) denotes the attention function [3], and W_v is a trainable weight matrix. By concatenating $\boldsymbol{s}_t$ with the attentive context vector $\boldsymbol{c}_t$ and entity vector $\boldsymbol{k}_t$, the model improves the ability of handling long-term dependency.

3.4. Training

Inspired by [9], we transform the system response $\boldsymbol{y}$ into sketch sequence $\boldsymbol{y}^s$ that excludes slot values but includes the slot tags. For example, the system response "*Valero is 1 mile away*" is transformed into "*@poi is @distance away*", where *@poi* and *@distance* are the slot tags that represent all the possible point of interest (POI) and distance values respectively. We estimate the parameters by minimizing negative log-likelihood of the entity and vocabulary distribution. Let $\boldsymbol{y}^s = (w_1^s, ..., w_n^s)$ be the sketch sequence, then loss function is defined as:

$$\mathcal{L} = -\sum_{t=1}^{n} (\text{log} P_t^{\text{vocab}}(w_t^s) + \text{log} P_t^{\text{entity}}(w_t^e)). \tag{17}$$

For the case where w_t is a non-entity token, we train the P_t^{entity} to produce a special token *@st*, i.e., $w_t^e =$ *@st*. In the decoding stage, we use a simple greedy strategy to generate the sketch sequence. If the generated word is a slot tag, we choose the entity with the highest probability of the entity distribution.

4. Experiments

4.1. Datasets

We evaluate our model on two public multi-turn task-oriented dialog datasets: InCar Assistant dataset [7] and CamRest dataset [19]. InCar includes three distinct domains: point-of-interest navigation (Nav), weather information retrieval (Wea), and calendar scheduling (Cal). For weather domain, we follow [20] to separate the highest temperature, lowest temperature and weather attribute into three different columns. For calendar domain, we ignore all the empty values for the case where there are some dialogs with a incomplete KB. CamRest dataset contains dialogs in restaurant reservation domain. We report our experimental results based on the relabeled version [12]. Especially, some human experts format the CamRest dataset by equipping the corresponding KB to every dialog. For both datasets, we detect entities in the dialog history with the global entity list provided by the datasets. The train/validation/test sets of these two datasets are split in advance by the providers. Table 1 summarizes the statistics of the datasets.

Table 1: *Statistics for two datasets.*

	InCar			CamRest
	Nav	Wea	Cal	
Vocabulary Size	1556			1164
KB Attribution Types	6	22	5	10
Train/Val/Test Dialogs	2425 / 302 / 304			406 / 135 / 136
Avg. Dialog Turns	2.3			5.1

4.2. Implementation Details

The model is trained end-to-end using Adam optimizer [21] with a fixed batch size of 8, and learning rate annealing starts from $1e^{-3}$ to $5 \times 1e^{-5}$. Embeddings of size 128 are randomly initialized and updated during training. We set the hidden size

Table 2: *Comparison of our model with baselines.*

	InCar					CamRest	
Model	BLEU	F1	Nav F1	Wea F1	Cal F1	BLEU	F1
Seq2Seq	11.58	23.81	28.55	25.95	13.37	17.20	45.83
Seq2Seq Attn	12.77	32.13	35.82	42.29	15.63	17.61	47.76
Mem2Seq	13.05	39.32	28.10	45.07	53.51	16.95	45.08
GLMP	14.76	57.73	50.55	**58.87**	69.05	18.74	53.24
Ours	**17.16**	**59.04**	**52.47**	57.78	**71.84**	**20.74**	**57.07**

of GRU to 128 for both encoder and decoder. The dropout rates are set to 0.4 for InCar and 0.5 for CamRest. We apply gradient clippling to 10 when its norm exceeds this value.

4.3. Baselines

We compare our model with following baseline models:

- **Seq2Seq** [2]: a standard seq2seq model with an encoder and a decoder. To incorporate the knowledge into generation, we implement the seq2seq model using the same training method as our model, i.e., the decoder is trained to generate entity distribution and vocabulary distribution (see more details in sec 3.4).
- **Seq2Seq Attn** [4]: Seq2Seq model with attention over the input context at each time step during decoding. Note that compared with our model, Seq2Seq Attn model uses learned embeddings as entity representation, while our model encodes entities with proposed EER.
- **Mem2Seq** [8]: Mem2Seq models dialog history and KB with a memory network and learns a pointer gate to control either generating a vocabulary word or selecting a word from input as the output.
- **GLMP** [9]: GLMP model adopts a global memory encoder and a local memory decoder to incorporate the shared external knowledge into the learning framework.

In our experiments, we train Seq2Seq and Seq2Seq Attn models with the same parameter settings as our model. For Mem2Seq and GLMP models, we run their official code with the suggested hyper-parameters.

4.4. Results

Following the prior works [8, 9], we adopt two automatic evaluation metrics to validate the performance of our model: BLEU [22] and Micro Entity F1. BLEU metric is commonly used to study the performance of task-oriented dialog models as it has been found to have strong correlation with huam judgements [23]. We compute the BLEU score using the `multi-bleu.perl` script from Moses. Entity F1 metric can evaluate the ability to generate relevant entities from the provided KB. To compute entity F1 score, we micro-average the precision and recall over the entire set of system responses. Since our model does not have slot-tracking by design, we evaluate on entity F1 instead of the slot-tracking accuracy as in [24].

In Table 2, we observe that our model achieves the highest BLEU and entity F1 scores on both datasets. Compared with model Seq2Seq Attn that learns word embeddings as entity representation, our model has about 27% and 10% entity F1 score improvement for InCar and CamRest respectively. Moreover, our model outperforms the previous state-of-the-art model GLMP in most domains, which further demonstrates the robustness of EER. In addition, our model has a significant improvement on BLEU, which shows that our model can effectively incorporate knowledge into dialog generation.

Table 3: *Results of OOV Test.*

	InCar		CamRest	
Model	BLEU	F1	BLEU	F1
Seq2Seq	10.35	15.10	16.56	37.95
Seq2Seq Attn	10.64	17.81	16.76	39.59
Mem2Seq	11.37	23.30	13.59	40.29
GLMP	12.92	50.86	16.80	50.07
Ours	**16.68**	**55.18**	**20.19**	**56.28**

4.5. OOV Test

To validate the ability of solving the common OOV problem, we conduct an OOV test where the entities that occur less frequently in the dataset are removed from vocabulary. We observe that most of the named entities from the KB are rare entities in the datasets. Thus, we first drop all the entities that belong to *point-of-interest*, *location*, *event* and *name* attributes from vocabulary, which corresponds to Nav, Wea, Cal domains of InCar and CamRest respectively. All these entities are then replaced by a special unknown token *unk*. In this way, we can alleviate the problem of the explosion in the vocabulary size and the number parameters since we only regard the most frequent entities in the training corpus. We train all the models with the new vocabulary of size 1458 for InCar and 1030 for CamRest.

In Table 3, We can see that Seq2Seq and Seq2Seq Attn models have significant performance drop on entity F1, which indicates that these two models rely on learned word embeddings for each entity. For Mem2Seq and GLMP models, they slightly mitigate the OOV problem since the multi-hop attention mechanisms of memory network help in learning correlations between memories [8]. Note that our model achieves the least OOV performance drop on both BLEU and entity F1 scores over all the compared models, which demonstrates the superiority of EER in handling OOV challenge.

5. Conclusion

In this paper, we propose a novel enhanced entity representation for end-to-end trainable task-oriented dialog systems. The model enhances entity representation with contextual and structural information from the dialog history and the provided KB. Moreover, we propose a switching network which enables the decoder to better incorporate the knowledge into dialog generation. Experimental results show that the proposed model can robustly represent entities, and outperforms existing state-of-the-art models on two automatic evaluation metrics. Furthermore, we conduct an OOV test to demonstrate the effectiveness of EER in dealing with the rare and unseen entities.

6. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems 27 (NIPS 2014)*, 2014, pp. 3104–3112.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014, pp. 1724–1734.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[4] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, 2015, pp. 1412–1421.

[5] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 280–290.

[6] M. Eric and C. D. Manning, "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 (EACL 2017)*, 2017, pp. 468–473.

[7] M. Eric and C. D. Manning, "Key-value retrieval networks for task-oriented dialogue," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2017)*, 2017, pp. 37–49.

[8] A. Madotto, C. Wu, and P. Fung, "Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018, pp. 1468–1478.

[9] C. Wu, R. Socher, and C. Xiong, "Global-to-local memory pointer networks for task-oriented dialogue," in *7th International Conference on Learning Representations (ICLR 2019)*, 2019.

[10] Z. Lin, X. Huang, F. Ji, H. Chen, and Y. Zhang, "Task-oriented conversation generation using heterogeneous memory networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, 2019, pp. 4557–4566.

[11] R. G. Reddy, D. Contractor, D. Raghu, and S. Joshi, "Multi-level memory for task oriented dialogs," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, 2019, pp. 3744–3754.

[12] L. Qin, Y. Liu, W. Che, H. Wen, Y. Li, and T. Liu, "Entity-consistent end-to-end task-oriented dialogue system with kb retriever," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, 2019, pp. 133–142.

[13] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Advances in neural information processing systems 28 (NIPS 2015)*, 2015, pp. 2440–2448.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS 2011)*, 2011, pp. 315–323.

[16] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, 2017.

[17] M. S. Schlichtkrull, T. Kipf, P. Bloem, R. V. Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference (ESWC 2018)*, 2018, pp. 593–607.

[18] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 2016, pp. 140–149.

[19] T. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojasbarahona, P. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, 2017, pp. 438–449.

[20] H. Wen, Y. Liu, W. Che, L. Qin, and T. Liu, "Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, 2018, pp. 3781–3792.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[22] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, 2002, pp. 311–318.

[23] S. Sharma, L. E. Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," *arXiv preprint arXiv:1706.09799*, 2017.

[24] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL 2014)*, 2014, pp. 263–272.