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

# End-to-End Task-oriented Dialog System through Template Slot Value Generation

*Teakgyu Hong*[1], *Oh-Woog Kwon*[2], *Young-Kil Kim*[2]
[1]Clova AI, NAVER Corp.
[2]Electronics and Telecommunications Research Institute
teakgyu.hong@navercorp.com, {ohwoog, kimyk}@etri.re.kr

## Abstract

To overcome the limitations of conventional pipeline-based task-oriented dialog systems, an end-to-end approach has been introduced. To date, many end-to-end task-oriented dialog systems have been proposed and these have shown good performance in various domains. However, those have some limitations such as the need for dialog state annotations. And there is also room for improvement for those systems. In this paper, we examine the issues of recent end-to-end task-oriented dialog systems and present a model that can handle these issues. The proposed model classifies a system utterance template in a retrieval-based manner and then generates the slot values in the template through a decoder. Also, we propose an unsupervised learning based template generation method that allows model training even in a domain where the templates are not given and the dialog information is not tagged. Our model obtains new state-of-the-art results on a restaurant search domain.
**Index Terms**: spoken dialog systems, task-oriented dialog, end-to-end model

## 1. Introduction

Task-oriented dialog systems help users to accomplish some goals using natural language. Conventional task-oriented dialog systems have been built as a pipeline, with modules for natural language understanding, dialog management, and natural language generation [1, 2]. However, those pipeline-structured systems were difficult to adapt to the new domain and, in those systems, the error occurred in the lower module can be propagated to the upper module [3]. To solve these problems, end-to-end methods have been proposed.

The end-to-end method was first applied to the chat-oriented dialog systems [4, 5, 6]. Based on the success of applying to the chat-oriented dialog systems, the end-to-end approach has also begun to be studied in task-oriented dialog systems. At first, recurrent neural network (RNN) or end-to-end memory network (MemN2N) [7] based task-oriented dialog models have been proposed. Then, based on these models, models which predict system utterances directly from dialog context and knowledge base (KB) search results have been presented. Recently, some models utilize not only system utterances, but also dialog states, dialog act, and so on in the end-to-end training phase. Also, pre-trained language models, such as Bidirectional Encoder Representations from Transformers (BERT) [8], have been used in recent end-to-end task-oriented dialog systems.

Among recently proposed end-to-end task-oriented dialog systems, some systems utilize dialog tagging information, such as a dialog state. Other systems directly generate a system utterance using a decoder. Those have achieved good performance in various domains. However, those have some limitations. For example, to utilize the dialog tagging information, this information must be tagged in the dialog corpus. Plus, some models have room for improvement. In the case of the models which generate a system utterance using a decoder, there is a big disadvantage that generating system utterances cannot be controlled, which is important in a task-oriented dialog system [9].

In this paper, we propose a model that can handle the limitations of the recently proposed end-to-end task-oriented dialog systems. The proposed model first classifies the system utterance template similar to the retrieval-based model, and then the slot values of the slot tags in the template are generated through a decoder. After that, the system utterance is generated by combining the classified template and the generated slot values. Also, we present a method for generating system utterance templates from a dialog corpus through unsupervised learning to enable training a model on corpora that do not provide system utterance templates. Experiments were conducted in a restaurant search domain, and the results show that our model achieved higher BLEU and Entity F1 scores than the other existing models.

## 2. Related Work

Since an end-to-end approach shows great results in chat-oriented dialog systems [4, 5, 6], this approach has also begun to be applied to the task-oriented dialog systems. When building end-to-end task-oriented dialog models, an RNN or MemN2N is mainly used to encode a dialog context.

Eric and Manning [10] framed a task-oriented dialog as a sequence-to-sequence (Seq2Seq) learning problem and built long short-term memory (LSTM) based encoder-decoder model to solve it. Williams et al. [9] introduced a hybrid code network (HCN), which is a dialog control model that combines RNN with domain-specific knowledge and system action templates. Liu and Lane [11] modeled task-oriented dialog as a multi-task sequence learning problem and implemented the system using a hierarchical RNN [12, 13].

Bordes et al. [14] suggested a method to encode dialog context using MemN2N and select system utterance. Madotto et al. [15] extended this method to decode system utterance using RNN. Raghu et al. [16] and Reddy et al. [17] proposed separate memories for encoding dialog context and KB search results when using MemN2N. In addition to these, there are studies on multiple answers [18], personalization [19], and multi-domain [20] based on MemN2N.

Recently, some models have been proposed to utilize intermediate dialog outputs, such as dialog state, system action, instead of using only system utterances [21, 22]. Also, dialog systems which are based on pre-trained language models have been presented [23, 21], since those have shown good performance in many natural language processing tasks [8, 24].



http://dx.doi.org/10.21437/Interspeech.2020-2011

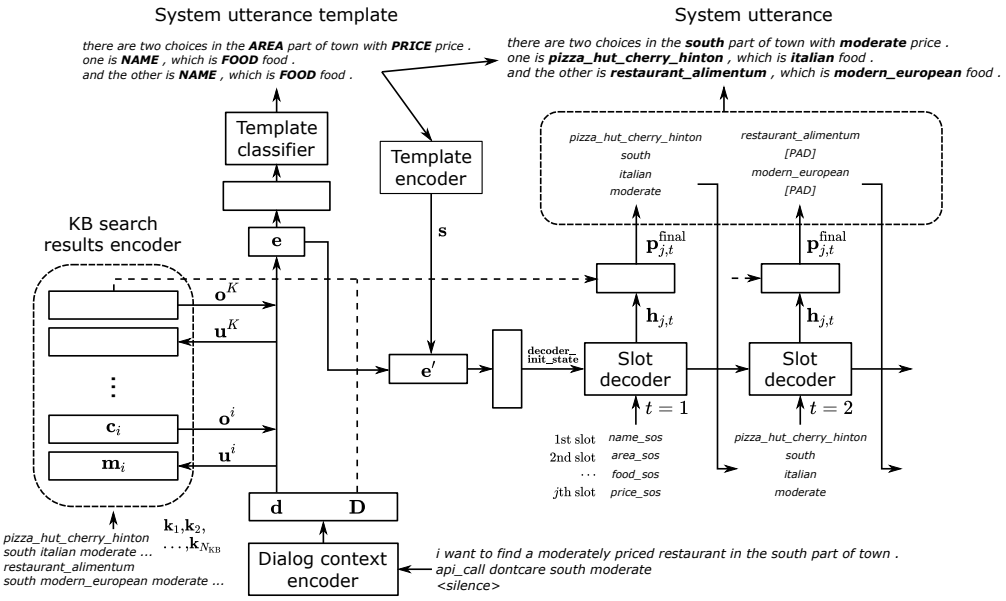


Figure 1: *The proposed end-to-end task-oriented dialog system through template slot value generation.*

# 3. Proposed Method

In this section, we analyze the limitations of existing end-to-end task-oriented dialog systems and present a model and method to overcome those limitations.

## 3.1. The Limitations of Existing Dialog Systems

There are two main characteristics of the recently proposed end-to-end task-oriented dialog systems. The first is that dialog labels, such as dialog state, are used when train a model. Some publicly available corpora provide such labels. However, most real dialog corpora do not have those kinds of information, and labeling such information requires a lot of human effort. Therefore, ultimately, it would be desirable to develop a model that can be trained without such tagging information.

The second is that the system utterance is generated in a word-by-word manner using a decoder. In this class of models, dialog context and KB search results are encoded, and then system utterance is generated through a decoder in a word-by-word manner. It has the advantage that any dialog labels are not required. However, since those are a generative model, problems of the generative model may occur as in chat-oriented dialog. Furthermore, those cannot control system utterances. Considering that the purpose of a task-oriented dialog system is to provide the user with desired information, it is important to be able to control the utterances. Plus, generative models have the problems of degenerated behavior and credit assignment over a long horizon when applying reinforcement learning [25].

## 3.2. End-to-End Task-oriented Dialog System through Template Slot Value Generation

We designed an end-to-end task-oriented dialog system to overcome the limitations mentioned in 3.1. The proposed model, shown in Figure 1, first classifies the system utterance template and then generates the slot values in the template through a decoder. The final system utterance is generated by combining the template and slot values. We introduce its detailed implementation in the below sections.

### *3.2.1. Encoding the dialog context and KB search results*

The dialog context indicates user and system utterances up to the current turn and the KB search results are the results of the system utterance requesting the KB search.

First, the dialog context is encoded through BERT. In BERT, there is a special token, *[CLS]*, used when pre-train the model. In our model, we regard the embedding of this token, $\mathbf{d} \in \mathbb{R}^{d_E}$, as a dialog context encoding, where $d_E$ is a hidden size of BERT. Then, to identify tokens related to the dialog context among the KB search results tokens, $\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_{N_{\text{KB}}} \in \mathbb{R}^{|V_{\text{KB}}|}$ (one-hot representations), where $N_{\text{KB}}$ is the maximum number of KB search results tokens, a query $\mathbf{u} \in \mathbb{R}^{d_M}$ is created based on the dialog context embedding $\mathbf{h}$, and then KB search results tokens are encoded through the MemN2N,

$$\mathbf{u} = \mathbf{d}\mathbf{M}_B \qquad \mathbf{m}_i = \mathbf{M}_A\mathbf{k}_i \quad \mathbf{c}_i = \mathbf{M}_C\mathbf{k}_i \quad (1)$$

$$p_i = \text{Softmax}(\mathbf{u}^\top \mathbf{m}_i) \qquad \mathbf{o}_i = p_i\mathbf{c}_i \qquad \mathbf{o} = \sum_i \mathbf{o}_i, \quad (2)$$

where $\mathbf{M}_B \in \mathbb{R}^{d_E \times d_M}$, $\mathbf{M}_A \in \mathbb{R}^{d_M \times |V_{\text{KB}}|}$, and $\mathbf{M}_C \in \mathbb{R}^{d_M \times |V_{\text{KB}}|}$ are embedding matrices, $d_M$ is a dimension of memory vector, and $|V_{\text{KB}}|$ is the size of KB vocabulary.

To extend our model to handle $K$ hop operations, we stack memory layers and explore two types of weight tying [7].

- Adjacent (Adj)
  - The input to layers above: $\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{o}^k$.
  - The output embedding for one layer is the input embedding for the one above, i.e. $\mathbf{M}_A^{k+1} = \mathbf{M}_C^k$.
- Layer-wise (Layer)
  - The input to layers above: $\mathbf{u}^{k+1} = \mathbf{G}\mathbf{u}^k + \mathbf{o}^k$, where $\mathbf{G} \in \mathbb{R}^{d_M \times d_M}$ is a linear mapping.
  - The input and output embeddings are the same across different layers, i.e. $\mathbf{M}_A^1 = \mathbf{M}_A^2 = ... = \mathbf{M}_A^K$ and $\mathbf{M}_C^1 = \mathbf{M}_C^2 = ... = \mathbf{M}_C^K$.

The final embedded representation of dialog context and KB search results is $\mathbf{e} = \mathbf{u}^{K+1} \in \mathbb{R}^{d_M}$.

#### 3.2.2. Classifying the system utterance template

From the dialog context and KB search results embedding $\mathbf{e}$, the system utterance template is classified through a softmax layer,

$$\mathbf{p}^{\text{sys}} = \text{Softmax}(\mathbf{W}_{\text{sys}}\mathbf{e}), \quad (3)$$

where $\mathbf{W}_{\text{sys}} \in \mathbb{R}^{N_{\text{sys}} \times d_M}$ is a trainable parameter and $N_{\text{sys}}$ is the number of templates.

#### 3.2.3. Decoding the slot values of the slot tags in the template

The slot values in the template will be affected by the classified template as well as the dialog context and KB search results. So, when setting the decoder initial state, we include both information sources,

$$\mathbf{e}' = \mathbf{e} \oplus \mathbf{s} \quad (4)$$

$$\textbf{decoder\_init\_state} = \tanh(\mathbf{W}_{\text{init}}\mathbf{e}' + \mathbf{b}_{\text{init}}), \quad (5)$$

where $\mathbf{s} \in \mathbb{R}^{N_{\text{sys}}}$ is the one-hot vector of template, $\oplus$ indicates concatenation, $\mathbf{W}_{\text{init}} \in \mathbb{R}^{d_D \times (d_M + N_{\text{sys}})}$ and $\mathbf{b}_{\text{init}} \in \mathbb{R}^{d_D}$ are weight and bias, respectively, and $d_D$ is a decoder state size. In training phase, $\mathbf{s}$ is set to ground truth and in inference phase, it is set to a template which has the highest value of $\mathbf{p}_{\text{sys}}$.

The slot values appearing in the template are mostly influenced by the dialog context and KB search results. So, when generating the slot values using a decoder, attention and copy mechanisms were applied to utilize that information. For $j$th slot tag, for each decoding step $t$, token probabilities through the attention mechanism are calculated as follows [10]:

$$\mathbf{X} = \mathbf{D} \oplus \mathbf{O} \quad (6)$$

$$\mathbf{u}_{j,t} = \tanh(\mathbf{X}\mathbf{W}_X + \mathbf{W}_H\mathbf{h}_{j,t})^\top \mathbf{v} \quad (7)$$

$$\mathbf{a}_{j,t} = \text{Softmax}(\mathbf{u}_{j,t}) \quad (8)$$

$$\tilde{\mathbf{x}}_{j,t} = \sum_i a_{j,t,i}\mathbf{x}_i \quad (9)$$

$$\mathbf{p}^{\text{attn}}_{j,t} = \text{Softmax}(\mathbf{W}_A[\mathbf{h}_{j,t} \oplus \tilde{\mathbf{x}}_{j,t}]) \quad (10)$$

where $\mathbf{D} \in \mathbb{R}^{N_{\text{con}} \times d_E}$ represents all dialog context tokens vectors of the final layer of BERT, $N_{\text{con}}$ is the maximum number of dialog context tokens, $\mathbf{O} \in \mathbb{R}^{N_{\text{KB}} \times d_M}$ represents all KB search result tokens vectors ($\{\mathbf{o}^K_i\}_{i=1}^{N_{\text{KB}}}$), $\mathbf{X} \in \mathbb{R}^{(N_{\text{con}}+N_{\text{KB}}) \times d_E}$ is a concatenation of dialog context and KB search results tokens embeddings, $\mathbf{x}_i \in \mathbb{R}^{d_E}$ is the $i$th row of $\mathbf{X}$, and $\mathbf{h}_{j,t} \in \mathbb{R}^{d_D}$ is a decoder hidden state. $\mathbf{W}_X \in \mathbb{R}^{d_E \times d_A}$, $\mathbf{W}_H \in \mathbb{R}^{d_A \times d_D}$, $\mathbf{v} \in \mathbb{R}^{d_A}$, and $\mathbf{W}_A \in \mathbb{R}^{|V| \times (d_D + d_E)}$ are trainable model parameters, where $V$ is vocabulary. In the $\tanh()$ of equation 7, the result of right-hand side broadcasts to the left-hand side for addition. For simplicity, we set $d_E = d_M = d_D$.

For $j$th slot tag, for each decoding step $t$, token probabilities through the copy mechanism are calculated as follows:

$$\mathbf{p}^{\text{con}}_{j,t} = \text{Softmax}(\mathbf{D}\mathbf{h}_{j,t}) \quad (11)$$

$$\mathbf{p}^{\text{KB}}_{j,t} = \text{Softmax}(\mathbf{O}\mathbf{h}_{j,t}). \quad (12)$$

Then, convert them to have $|V|$ dimension, $\tilde{\mathbf{p}}^{\text{con}}_{j,t}, \tilde{\mathbf{p}}^{\text{KB}}_{j,t} \in \mathbb{R}^{|V|}$, and combine with a soft gating mechanism

$$\alpha^{\text{con-KB}}_{j,t} = \text{Sigmoid}(\mathbf{w}_C[\mathbf{h}_{j,t} \oplus \mathbf{d} \oplus \mathbf{o}^K]) \quad (13)$$

$$\mathbf{p}^{\text{con-KB}}_{j,t} = \alpha^{\text{con-KB}}_{j,t}\tilde{\mathbf{p}}^{\text{con}}_{j,t} + (1 - \alpha^{\text{con-KB}}_{j,t})\tilde{\mathbf{p}}^{\text{KB}}_{j,t}, \quad (14)$$

where $\mathbf{w}_C \in \mathbb{R}^{d_D + d_E + d_M}$ is a trainable parameter.

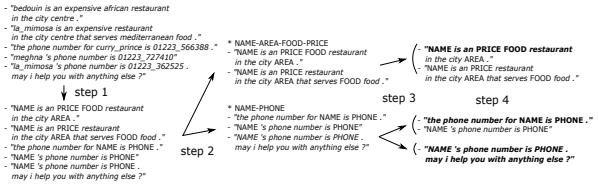


Figure 2: *Unsupervised learning based template generation. Utterances in bold are used as templates.*

To generate a token from attention and copy probabilities, we once again combine with a soft gating mechanism

$$\alpha^{\text{final}}_{j,t} = \text{Sigmoid}(\mathbf{w}_F[\mathbf{h}_{j,t} \oplus \mathbf{d} \oplus \mathbf{o}^K]) \quad (15)$$

$$\mathbf{p}^{\text{final}}_{j,t} = \alpha^{\text{final}}_{j,t}\mathbf{p}^{\text{attn}}_{j,t} + (1 - \alpha^{\text{final}}_{j,t})\mathbf{p}^{\text{con-KB}}_{j,t}, \quad (16)$$

where $\mathbf{w}_F \in \mathbb{R}^{d_D + d_E + d_M}$ is a trainable parameter. The slot value token is generated from $\mathbf{p}^{\text{final}}_{j,t}$.

#### 3.2.4. Training the model

The model is trained to minimize the sum of cross-entropy losses for template and slot value tokens

$$\min_\theta \sum_{i=1} -[\log \mathbf{p}^{\text{sys}}(S^*_i; \theta) + \sum_{j=1}\sum_{t=1} \log \mathbf{p}^{\text{final}}_{j,t}(y^*_{i,j,t}; \theta)], \quad (17)$$

where $\theta$ is a parameter set and $S^*_i$ and $y^*_{i,j,t}$ are ground truth template and slot value token, respectively.

### 3.3. Unsupervised Learning based Template Generation

Most dialog corpora collected by Wizard-of-Oz method have various forms of utterances. If we delexicalize utterances with ontology and KB by string matching and use all delexicalized utterances as templates, the number of templates becomes too large and it leads to bad performance. Therefore, we propose the following 4 steps of unsupervised learning based template generation, which is shown in Figure 2, to obtain compact templates while controlling the number of them.

- 1) Delexicalize utterances with ontology and KB by string matching
- 2) Group utterances based on the slot tag combination
- 3) Perform clustering on utterances in the same group
- 4) Choose utterance that appears most frequently among the utterances in the cluster as a template

In step 3, we vectorize utterances through Universal Sentence Encoder [26] and then apply hierarchical clustering since it showed the best performance in dialog act clustering [27].

### 3.4. Advantages of the Proposed Model

There are 4 main advantages of the proposed model.

- 1) Template-based: It can control system utterance.
- 2) BERT-based: It can utilize representations obtained by BERT and handle unseen and unknown slot values by using subword vocabulary.
- 3) Implicit dialog state tracking (DST): It performs DST implicitly by generating slot values. It helps the model to understand dialog better.
- 4) Learning how to handle KB: It can retrieve related tokens from the KB without relying on string matching which cannot capture semantically the same, but different in shape tokens (e.g. *"moderately"* vs. *"moderate"*).

| Weight tying | # hops | Metric | # epochs | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 40 | 48 | 56 | 64 | 72 | 80 |
| Adj | 3 | BLEU | 14.8 | 15.7 | 15.8 | 16.7 | **16.4** | 17.0 |
| | | F1 | 58.4 | 58.9 | 60.5 | 61.6 | **62.2** | 62.0 |
| | 6 | BLEU | 14.3 | 15.3 | 15.8 | 16.4 | 16.9 | 16.3 |
| | | F1 | 57.3 | 58.3 | 59.8 | 60.4 | 62.1 | 61.8 |
| Layer | 3 | BLEU | 13.8 | 15.3 | 16.0 | 16.0 | 16.6 | 16.8 |
| | | F1 | 57.0 | 59.1 | 60.4 | 60.9 | 61.9 | 61.6 |
| | 6 | BLEU | 13.7 | 14.8 | 15.2 | 15.9 | 16.0 | 16.2 |
| | | F1 | 56.7 | 57.2 | 59.8 | 60.8 | 61.4 | 61.3 |

Table 1: *Experimental results according to various settings. Scores are the average of 5 experiments.*

| Model | BLEU | Entity F1 |
|---|---|---|
| Attn seq2seq [30] | 7.7 | 25.3 |
| Ptr-UNK [31] | 5.1 | 40.3 |
| KVRet [32] | 13.0 | 36.5 |
| Mem2Seq [15] | 14.0 | 52.4 |
| MM [17] | 15.9 | 61.4 |
| Ours | **16.4** $\pm$ 0.2 | **62.2** $\pm$ 0.4 |

Table 2: *Comparision with other models. Scores are the average and standard error of 5 experiments.*

| | |
|---|---|
| Case 1 | |
| Context | User: "hello , i am looking for an expensive restaurant that serves australian food ." |
| KB | |
| GT | "there are no records matching your request for an expensive restaurant serving australian food . may i help you find a different restaurant ?" |
| Ours | **"i 'm sorry , there are no expensive restaurants that serve australian food . can i help you with something else ?"** |
| Case 2 | |
| Context | User: "is there a cheap restaurant serving belgian food ?"<br>System: "what area are you interested in ?"<br>User: "i do not care what area it is in ."<br>System: "there are no cheap belgian restaurants . would you like to change your query ?"<br>User: "how about a cheap restaurant serving italian food ?"<br>System: "api_call italian dontcare cheap"<br>User: "<silence>" |
| KB | ["la_margherita", "west", "italian", "cheap", ...],<br>["da_vinci_pizzeria", "north", "italian", "cheap", ...],<br>["ask", "centre", "italian", "cheap", ...],<br>... |
| GT | "da_vinci_pizzeria serves italian food in the cheap price range and in the centre of town , is there anything else i can help you with ?" |
| Ours | "for which would you like more information , the **la_margherita** , or the **da_vinci_pizzeria** ." |
| Case 3 | |
| Context | User: "i am looking for a restaurant that is in the expensive price range and in the south part of town ."<br>System: "there are results matching your query . would you like mexican , italian , chinese , or indian ?"<br>User: "let 's go with italian food ."<br>System: "api_call italian south expensive"<br>User: "<silence>"<br>System: "frankie_and_bennys is an expensive italian eatery in the south part of town . would you like any additional information about this restaurant ?"<br>User: "address and phone number" |
| KB | ... |
| GT | "the address is cambridge_leisure_park_clifton_way cherry hinton . the phone number is 01223_412430 ." |
| Ours | "**NAME** is located at **01223_412430** and the phone number is **01223_412430** ." |

Table 3: *Comparison between ground truth (GT) system utterances and utterances generated by the proposed model.*

# 4. Experiments

## 4.1. Dataset

We use a restaurant search domain, CamRest [28], to evaluate our model. It consists of 406, 135, and 135 dialogs for Train, Dev, and Test set, respectively, and there are 7 slots in total, including 3 informable slots (area, food, price range). Since the experimental results on this dataset are different slightly from paper to paper due to differences in preprocessing, we used a dataset preprocessed and released by Reddy et al. [17].

## 4.2. Experimental Settings

We perform mini-batch training with batch size 8 using Adam optimizer [29]. The initial learning rate was set to 5e-5 and the number of training epochs was tested up to 80. The maximum sequence length was set to 64 for both BERT and MemN2N.

The use of subwords has the advantage of being able to respond to unseen and unknown slot values, but some words within the domain were over tokenized (e.g. *"don_pasquale_pizzeria"* $\rightarrow$ [*"don"*, *"_"*, *"pas"*, *"##qual"*, *"##e"*, *"_"*, *"pi"*, *"##zz"*, *"##eria"*]). Such excessive tokenization decreases the number of tokens that can be encoded and degrades performance since the number of decoding steps increases. To bypass this problem, we replaced *[unusedXX]* tokens, which are not used but exist in BERT vocabulary, with domain words to prevent excessive tokenization while utilizing the pre-trained BERT.

## 4.3. Results and Analysis

Table 1 shows the experimental results with different types of weight tyings and the number of hops and training epochs. In Table 1, BLEU measures the n-gram overlap with the ground truth system utterance, and Entity F1 is the F1 score of slot value entities appearing in the utterance [17]. It shows that adjacent weight tying achieved slightly better results than layer-wise, and as the number of training epochs increases, performance tends to increase and then converge. Also, the number of hops didn't affect the score as the number of epochs is increased. The scores written in bold in Table 1 were the results that achieved the highest BLEU + Entity F1 in the Dev set.

In Table 2, we compare our model with other existing models. The results of other models are reported from Reddy et al. [17]. Our model achieved state-of-the-art results in both scores.

## 4.4. The Effect of MemN2N based KB Encoding

To investigate the ability to handle KB search results through the MemN2N, we removed MemN2N and modified the model to encode dialog context and KB search results, respectively, through two different BERT models. In the same setting in Table 2, BLEU and Entity F1 decrease from 16.4 to 13.7 and from 62.2 to 53.6, for each. From this result, it can be seen that independent encoding cannot focus on KB search results related to the dialog context despite the attention and copy mechanisms. Therefore when encoding KB search results, it seems essential to explicitly reflect the dialogue context similar to our model.

## 4.5. Utterances Generated by the Model

Table 3 compares the ground truth system utterances and utterances generated by the proposed model. In Case 1, the model selected a semantically similar template and decoded the proper slot values. So, it generated an appropriate system utterance, even if it is not the same as the ground truth. In Case 2, the model decoded multiple slot values for one slot tag, *NAME*. However, in Case 3, *NAME* slot fails to be decoded and a slot value of *PHONE* slot is decoded in the position of *ADDRESS* slot. This seems to be due to all slots sharing one decoder.

# 5. Conclusions

In this paper, we propose a model that can overcome the limitations of the existing end-to-end task-oriented dialog systems. Also, we present an unsupervised learning based template generation method. Our model achieved state-of-the-art results in a restaurant search domain and we show that it can generate appropriate system utterances.

# 6. Acknowledgements

This work was supported by IITP grant funded by the Korea government (MSIT) (2019-0-00004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners).

# 7. References


[1] A. I. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, and A. Oh, "Creating natural dialogs in the carnegie mellon communicator system," in *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, 1999.

[2] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[3] T. Zhao and M. Eskenazi, "Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2016, pp. 1–10.

[4] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2015, pp. 1577–1586.

[5] O. Vinyals and Q. Le, "A neural conversational model," in *Proceedings of the ICML 2015 Deep Learning Workshop*, 2015.

[6] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NA-HLT)*, 2015, pp. 196–205.

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

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019, pp. 4171–4186.

[9] J. D. Williams, K. Asadi, and G. Zweig, "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 (ACL)*, 2017, pp. 665–677.

[10] M. Eric and C. 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, pp. 468–473.

[11] B. Liu and I. Lane, "An end-to-end trainable neural network model with belief tracking for task-oriented dialog," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2506–2510.

[12] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, 2015, pp. 553–562.

[13] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[14] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[15] A. Madotto, C.-S. 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, pp. 1468–1478.

[16] D. Raghu, N. Gupta *et al.*, "Disentangling language and knowledge in 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, pp. 1239–1255.

[17] 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, pp. 3744–3754.

[18] J. Rajendran, J. Ganhotra, S. Singh, and L. Polymenakos, "Learning end-to-end goal-oriented dialog with multiple answers," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 3834–3843.

[19] L. Luo, W. Huang, Q. Zeng, Z. Nie, and X. Sun, "Learning personalized end-to-end goal-oriented dialog," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, 2019, pp. 6794–6801.

[20] L. Qin, X. Xu, W. Che, Y. Zhang, and T. Liu, "Dynamic fusion network for multi-domain end-to-end task-oriented dialog," 2020.

[21] D. Ham, J.-G. Lee, W. Jang, and K.-E. Kim, "End-to-end neural architecture for pipelined dialogue system using gpt-2," in *Proceedings of the AAAI 2020 DSTC8 Workshop*, 2020.

[22] W. Liang, Y. Tian, C. Chen, and Z. Yu, "Moss: End-to-end dialog system framework with modular supervision," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[23] P. Budzianowski and I. Vulić, "Hello, its gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 15–22.

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[25] T. Zhao, K. Xie, and M. Eskenazi, "Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACCL-HLT)*, 2019, pp. 1208–1218.

[26] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," 2018.

[27] C. Shi, Q. Chen, L. Sha, S. Li, X. Sun, H. Wang, and L. Zhang, "Auto-dialabel: Labeling dialogue data with unsupervised learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 684–689.

[28] T.-H. Wen, M. Gasic, N. Mrkšić, L. M. R. Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young, "Conditional generation and snapshot learning in neural dialogue systems," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2153–2162.

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

[30] M.-T. 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, pp. 1412–1421.

[31] 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, pp. 140–149.

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