

Dual Dynamic Memory Network for End-to-End Multi-turn Task-oriented Dialog Systems

Jian Wang¹ Junhao Liu² Wei Bi Xiaojiang Liu
Kejing He^{1*} Ruifeng Xu³ Min Yang^{2*}

¹South China University of Technology

²Shenzhen Key Laboratory for High Performance Data Mining,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

³Harbin Institute of Technology (Shenzhen)

cs.wangjian@mail.scut.edu.cn, {jh.liu, min.yang}@siat.ac.cn
{victoriabi, kieranliu}@gmail.com
kjhe@scut.edu.cn, xuruifeng@hit.edu.cn

Abstract

Existing end-to-end task-oriented dialog systems struggle to dynamically model long dialog context for interactions and effectively incorporate knowledge base (KB) information into dialog generation. To conquer these limitations, we propose a Dual Dynamic Memory Network (DDMN) for multi-turn dialog generation, which maintains two core components: dialog memory manager and KB memory manager. The dialog memory manager dynamically expands the dialog memory turn by turn and keeps track of dialog history with an updating mechanism, which encourages the model to filter irrelevant dialog history and memorize important newly coming information. The KB memory manager shares the structural KB triples throughout the whole conversation, and dynamically extracts KB information with a memory pointer at each turn. Experimental results on three benchmark datasets demonstrate that DDMN significantly outperforms the strong baselines in terms of both automatic evaluation and human evaluation. Our code is available at <https://github.com/siat-nlp/DDMN>.

1 Introduction

Task-oriented dialog systems are designed to help users achieve specific goals with natural language, such as weather inquiry or restaurant reservation. Compared with traditional pipeline methods (Williams and Young, 2007; Young et al., 2013), end-to-end approaches recently have gained much attention (Zhao et al., 2017; Eric and Manning, 2017; Madotto et al., 2018), since they free the task-oriented dialog systems from the manually designed pipeline modules and can be automatically scaled up to new domains.

Recently, sequence-to-sequence (Seq2Seq) models have dominated the study of end-to-end task-oriented dialog systems (Bordes et al., 2017). Different from typical Seq2Seq models for open-domain dialog systems, the successful conversations for task-oriented dialog systems heavily rely on both dialog history and domain-specific knowledge base (KB). To effectively incorporate KB information and perform knowledge-based reasoning, memory augmented models have been proposed (Madotto et al., 2018; Wu et al., 2019), which model the dialog history and the KB knowledge as a bag of words in a flat memory.

Despite the remarkable progress of previous studies, current memory based models for multi-turn task-oriented dialog systems still suffer from the following limitations. **First**, existing methods concatenate dialog utterances of current turn and previous turns as a whole, which ignore previous reasoning process performed by the model and are incapable of dynamically tracking long-term dialog states. These methods introduce much noise since previous utterances as the context is lengthy and redundant (Zhang et al., 2018). Taking the dialog in Table 1 as an example, when answering the user question in 6-th turn, it is difficult for the model to infer that the name of the restaurant is “cocum” from a long concatenated dialog context. Therefore, previous models struggle to work well in the situations that require many rounds of interactions to complete a specific task. **Second**, previous studies tend to confound dialog history with KB knowledge, and store them into a flat memory (Sukhbaatar et al., 2015; Eric and Manning, 2017; Madotto

*Corresponding authors. This work was conducted when Jian Wang was interning at SIAT, Chinese Academy of Sciences. This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

Subject	Area	Address	Food	Price
cocum	west	71_castle_street_city_centre	indian	expensive
rice_boat	west	37_newnham_road_newham	indian	expensive
la_tasca	centre	14-16_bridge_street	spanish	moderate
...

Role	Turn	Utterance
User	1	hello, i am looking for a scottish restaurant on the west side of town.
System	1	i can help you with that. it looks like there are not any restaurants like that in my database. can i help you find a different restaurant?
User	2	is there a restaurant that serves indian food in the west part of town?
System	2	i found several indian restaurants in the west part of town. were you looking for an expensive or moderately priced restaurant?
User	3	expensive.
System	3	i found several in the expensive price range in the west part of town, can i help you with anything else?
User	4	can you please give me that phone number?
System	4	cocum is the first expensive indian restaurant in the west part of town on the database. the phone number for cocum is 01223 366668.
User	5	what is the food type they serve?
System	5	they serve indian food.
User	6	can you please give me the address?
System	6	cocum is located at 71_castle_street_city_centre. is there anything else i can help you with?
User	7	no, thank you, that is all i need. goodbye!
System	7	thank you. good day!

Table 1: A restaurant reservation example based on the CamRest dataset along with the KB information.

et al., 2018). The shared memory suffers from encoding the dialog context and KB information using a single strategy, which makes it hard to efficiently reason over the two different types of data, especially when the memory is large.

To alleviate the aforementioned limitations, we propose a Dual Dynamic Memory Network (DDMN), which keeps track of the long-term dialog history and KB knowledge with separate memories. Specifically, we leverage a dialog memory manager to effectively maintain history utterances with a dialog history memory and a dialog state memory. The dialog history memory keeps fixed to store the representation of dialog context throughout the whole conversation, and the dialog state memory keeps updated at each turn to track the flow of history information and capture proper information of current turn for generation. We leverage a KB memory manager containing a KB memory and a KB memory pointer to effectively track KB knowledge. The KB memory stores the KB triples using an end-to-end memory network (Sukhbaatar et al., 2015) and is shared across the entire conversation. The KB memory pointer softly attends to the KB memory at each turn, and guides the model to select appropriate KB entries in decoding.

Our main contributions can be summarized as follows.

- We propose a Dual Dynamic Memory Network (DDMN) for task-oriented dialog systems, which dynamically keeps track of long dialog context for multi-turn interactions and effectively incorporates KB knowledge into generation.
- We employ separate memories to model dialog context and KB triples. The iterative interactions between the two kinds of memories make the decoder focus on relevant dialog context and KB facts for generating coherent and human-like dialogs.
- The experimental results on three public datasets show that DDMN achieves impressive results compared to the existing methods. More importantly, our model is able to maintain more sustained conversations than the compared methods with the increase of dialog turns.

2 Model Description

Let $D = \{d_i = \langle u_i, s_i \rangle\}_{i=1}^M$ denote a set of dialogs, and M is the number of dialog turns. u_i and s_i denote the user utterances and system responses, respectively. Given a context X with utterances $\{d_j\}_{j=1}^{m-1}$ and u_m , a sequence of KB triples $B = \{b_1, b_2, \dots, b_l\}$, where m denotes the turn of current dialog, l is the number of KB triples, each triple is composed of $\langle \text{subject}, \text{relation}, \text{object} \rangle$. The objective of task-oriented dialog generation is to generate a proper response $Y = \{y_1, y_2, \dots, y_n\}$ word by word.

As shown in Figure 1, our proposed DDMN architecture consists of four components: a dialog encoder, a dialog memory manager, a KB memory manager, and a decoder. We elaborate on the proposed model in

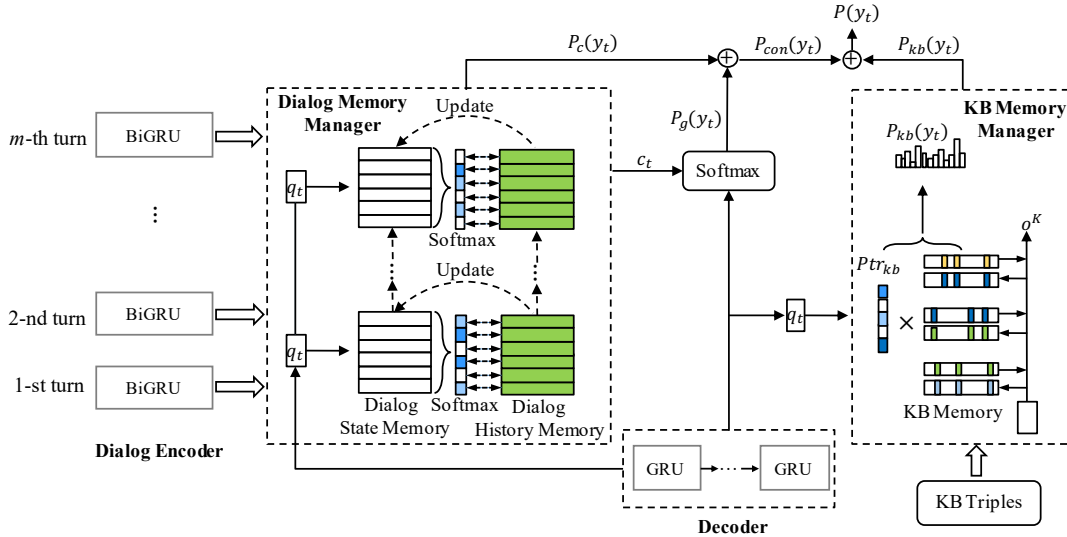


Figure 1: Overview of the Dual Dynamic Memory Network (DDMN) architecture.

detail as below.

2.1 Dialog Encoder

To overcome the challenge of modeling long dialog context in multi-turn conversations, the dialog encoder encodes dialog history utterances turn by turn. Specifically, for the first turn, the input to the encoder is u_1 . For j -th ($j > 1$) turn, the input is $\{s_{j-1}, u_j\}$, which is the concatenation of system response of the previous turn and user utterance of the current turn. Concretely, the input of dialog encoder at each turn are a sequence of tokens $x = (x_1, x_2, \dots, x_n)$, where n is the number of tokens. We first convert each token into a word vector through a randomly initialized trainable embedding matrix, and then employ a bidirectional gated recurrent unit (BiGRU) (Chung et al., 2014) to encode the input into hidden states:

$$\mathbf{h}_t = \text{BiGRU}(\mathbf{e}(x_t), \mathbf{h}_{t-1}) \quad (1)$$

where $\mathbf{e}(x_t)$ is the embedding of the token x_t . We take the concatenation of the forward and backward hidden states as the the output of the encoder, denoted as $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_n)$, which is then passed into the dialog memory manager for global management.

2.2 Dialog Memory Manager

The dialog memory manager maintains a dialog history memory and a dialog state memory, which are both initialized with the encoder hidden states of the first turn. For j -th ($j > 1$) turn, both the dialog history memory and the dialog state memory are “expanded” by concatenating the hidden states of j -th turn. The two memories are maintained throughout the whole conversation, with the dialog history memory keeping fixed to store the representation of dialog context of all previous turns. The dialog state memory keeps updated at each turn, which aims to track the flow of history information and capture proper information of current turn for response generation.

Generally, the decoder applies a GRU network to generate response word by word. At step t , the decoder state \mathbf{s}_t can be updated by:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, \mathbf{e}(y_{t-1})) \quad (2)$$

where $\mathbf{e}(y_{t-1})$ is the embedding of the previous word y_{t-1} . Here, \mathbf{s}_t is regarded as a “query” vector \mathbf{q}_t , which is used to attend to the dialog state memory and obtain the weighted context representation \mathbf{c}_t by reading from the dialog history memory. Then the dialog state memory will be updated with \mathbf{q}_t and \mathbf{c}_t by R rounds. Formally, let $\mathbf{K} \in \mathbb{R}^{N \times d}$ and $\mathbf{V} \in \mathbb{R}^{N \times d}$ be the dialog state memory and dialog history memory respectively, where N is the number of the memory slots and d is the dimension of vector in each slot, the detailed memory updating operations at round r ($r \in [1, R]$) are introduced as below.

Dialog State Memory Addressing The addressing operation aims to specify the normalized weights assigned to memory slots in $\mathbf{K}^{(r-1)}$ (the dialog state memory at $r - 1$ -th round), which formulates an

attention vector $\tilde{\mathbf{a}}_t^{(r)}$ at time step t . The j -th value of $\tilde{\mathbf{a}}_t^{(r)}$ is given by:

$$\tilde{a}_{t,j}^{(r)} = \text{softmax}(e_{t,j}), \quad e_{t,j} = \mathbf{v}_a^{(r)T} \tanh(\mathbf{W}_a^{(r)} \mathbf{q}_t + \mathbf{U}_a^{(r)} \mathbf{k}_{t,j}^{(r-1)}) \quad (3)$$

where $\mathbf{v}_a^{(r)}$, $\mathbf{W}_a^{(r)}$ and $\mathbf{U}_a^{(r)}$ are learnable parameters, $\mathbf{k}_{t,j}^{(r-1)}$ is the j -th slot in $\mathbf{K}^{(r-1)}$ at time step t .

Dialog History Memory Reading The reading operation reads from the dialog history memory \mathbf{V} to get the context representation $\tilde{\mathbf{c}}_t^{(r)}$ with the guidance of $\tilde{\mathbf{a}}_t^{(r)}$. The output of reading is given by $\tilde{\mathbf{c}}_t^{(r)} = \sum_{j=1}^N \tilde{a}_{t,j}^{(r)} \mathbf{v}_j$, where \mathbf{v}_j is the j -th memory slot in \mathbf{V} .

Dialog State Memory Updating Inspired by the read-write operations (Meng et al., 2016; Meng et al., 2018), we define two types of operations for updating the dialog state memory: FORGET and ADD. FORGET is analogous to the forget gate in GRU, which determines the information to be removed from memory slots. Similarly, ADD operation decides how much current information should be written to the dialog state memory as the added content.

Specifically, we first deploy another GRU network to imitate the decoder at round r , and obtain the ‘‘intermediate’’ hidden state $\tilde{\mathbf{s}}_t^{(r)}$ with reading output:

$$\tilde{\mathbf{s}}_t^{(r)} = \text{GRU}(\mathbf{q}_t, \tilde{\mathbf{c}}_t^{(r)}) \quad (4)$$

where $\tilde{\mathbf{s}}_t^{(r)}$ is used to update the dialog state memory. Then, the ‘‘intermediate’’ dialog state memory after FORGET operation is given by:

$$\tilde{\mathbf{k}}_{t,i}^{(r)} = \mathbf{k}_{t,i}^{(r-1)} (1 - a_{t,i}^{(r)} \cdot \mathbf{F}_t^{(r)}), \quad \mathbf{F}_t^{(r)} = \text{Sigmoid}(\mathbf{W}_F^{(r)} \tilde{\mathbf{s}}_t^{(r)}) \quad (5)$$

where the computation of $a_{t,i}^{(r)}$ is similar to that of $\tilde{a}_{t,j}^{(r)}$ defined in Eq.(3), $\mathbf{W}_F^{(r)} \in \mathbb{R}^{d \times d}$ is a learnable parameter. The dialog state memory after ADD operation is given by:

$$\mathbf{k}_{t,i}^{(r)} = \tilde{\mathbf{k}}_{t,i}^{(r)} + a_{t,i}^{(r)} \cdot \mathbf{A}_t^{(r)}, \quad \mathbf{A}_t^{(r)} = \text{Sigmoid}(\mathbf{W}_A^{(r)} \tilde{\mathbf{s}}_t^{(r)}) \quad (6)$$

where $\mathbf{W}_A^{(r)} \in \mathbb{R}^{d \times d}$ is a learnable parameter.

After R rounds of updating, the dialog state memory is modified with FORGET and ADD operations. Due to the ‘‘expansion’’ of dialog state/history memory along with the increase of the dialog turns, the dialog memory manager is able to dynamically keep track of long-term dialog state.

2.3 KB Memory Manager

To incorporate external knowledge effectively, the KB memory manager adopts end-to-end multi-hop memory networks (MemNN) (Sukhbaatar et al., 2015) for encoding structural KB information. Given the KB triples $B = \{b_1, b_2, \dots, b_l\}$, each entry $b_i \in B$ is represented in the format of a triple $\langle \text{subject}, \text{relation}, \text{object} \rangle$. The KB memory is then represented as a set of trainable embedding matrices $\mathbf{C} = (\mathbf{C}^1, \dots, \mathbf{C}^{K+1})$ and $\mathbf{C}^k \in \mathbb{R}^{V \times d}$ using MemNN, where K is the number of memory hops and V is the vocabulary size of the KB. It is noteworthy that our KB memory is shared across the entire conversation. Formally, an initial query vector \mathbf{q}^1 is used as the reading head, and it loops over K hops and computes the attention weights at each hop k as:

$$p_i^k = \text{softmax}((\mathbf{q}^k)^T \mathbf{c}_i^k) \quad (7)$$

where $\mathbf{c}_i^k = \mathbf{C}^k(b_i) \in \mathbb{R}^d$ is the memory content in i -th position. Note that $\mathbf{p}^k \in \mathbb{R}^l$ is a soft memory attention that decides the memory relevance with respect to the query vector. Then the KB memory manager reads the memory \mathbf{o}^k by the weighted sum over \mathbf{c}^{k+1} and update the query vector \mathbf{q}^{k+1} as:

$$\mathbf{o}^k = \sum_i p_i^k \mathbf{c}_i^{k+1}, \quad \mathbf{q}^{k+1} = \mathbf{q}^k + \mathbf{o}^k. \quad (8)$$

To further strengthen the ability of selecting correct KB entries, we introduce a KB memory pointer Ptr_{kb} , inspired by Wu et al. (2019). Note that the proposed pointer Ptr_{kb} is passed to the decoder turn by turn. Suppose Ptr_{kb} is denoted as a sequence of pointers $(\sigma_1, \sigma_2, \dots, \sigma_l)$, each pointer is formulated by:

$$\sigma_i = \text{Sigmoid}((\mathbf{q}^K)^T \mathbf{c}_i^K) \quad (9)$$

where \mathbf{q}^K and \mathbf{c}_i^K are the query vector and the memory content at the last hop, respectively. We add an auxiliary classification task to train Ptr_{kb} . We first define the corresponding label $Ptr^{label} = (g_1, g_2, \dots, g_l)$ by checking whether the *object* words in the KB memory exist in the expected system

response Y , where $g_i = 1$ if $object(b_i) \in Y$, otherwise $g_i = 0$. Then the KB memory pointer is trained using binary cross-entropy loss:

$$Loss_p = - \sum_{i=1}^l g_i \log \sigma_i + (1 - g_i) \log (1 - \sigma_i) \quad (10)$$

2.4 Decoder

The decoder generates a response word by word. In particular, a word at time step t is either generated from the vocabulary or copied from one of the two memories (dialog history memory or KB memory). First, the decoder employs a GRU network defined in Eq. (2) for generation. The generation distribution over vocabulary $P_g(y_t)$ can be obtained by feeding the decoder state \mathbf{s}_t and \mathbf{c}_t (the reading output of dialog history memory at the last round) into a softmax layer, which is given by

$$P_g(y_t) = \text{softmax}(\mathbf{W}_1[\mathbf{s}_t; \mathbf{c}_t]) \quad (11)$$

where \mathbf{W}_1 is a trainable parameter.

Second, following the copy mechanism (Gulcehre et al., 2016) that the attention scores are viewed as the probability to form the copy distribution, we adopt the addressing result of dialog state memory at the last round as the attention score $a_{t,j}$, thus the copy distribution over the dialog history memory is given by $P_c(y_t = w) = \sum_{tj:w_{tj}=w} a_{t,j}$.

Third, we use the KB memory pointer $P_{tr_{kb}}$ to dynamically access the KB memory, and then employ the decoder state \mathbf{s}_t defined in Eq. (2) to attend over the KB memory:

$$\beta_{t,j} = \text{softmax}(\alpha_{tj}), \quad \alpha_{tj} = \mathbf{v}_b^T \tanh(\mathbf{W}_b \mathbf{s}_t + \mathbf{U}_b \tilde{\mathbf{c}}_j^K), \quad \tilde{\mathbf{c}}_j^K = \mathbf{c}_j^K \times \sigma_j, \quad j \in [1, l] \quad (12)$$

where $\mathbf{v}_b, \mathbf{W}_b, \mathbf{U}_b$ are parameters to be learned, \mathbf{c}_j^K is the KB memory content in j -th position at the last hop. Therefore, the copy distribution over the KB memory is given by $P_{kb}(y_t = w) = \sum_{tj:w_{tj}=w} \beta_{t,j}$. Note that we copy the *object* word once a KB memory position (i.e., a KB triple $\langle \text{subject}, \text{relation}, \text{object} \rangle$) is pointed to.

We use a soft gate g_1 to control whether a word is generated from vocabulary or copied from dialog history memory by combining $P_g(y_t)$ and $P_c(y_t)$:

$$g_1 = \text{Sigmoid}(\mathbf{W}_2[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_2), \quad P_{con}(y_t) = g_1 P_g(y_t) + (1 - g_1) P_c(y_t) \quad (13)$$

Moreover, we use another gate g_2 to obtain the final output distribution $P(y_t)$ by leveraging $P_{kb}(y_t)$ and $P_{con}(y_t)$:

$$g_2 = \text{Sigmoid}(\mathbf{W}_3[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_3), \quad P(y_t) = g_2 P_{kb}(y_t) + (1 - g_2) P_{con}(y_t) \quad (14)$$

Therefore, the decoder loss is the cross-entropy between the output distribution $P(y_t)$ and the reference distribution p_t , denoted as $Loss_d = - \sum p_t \log(P(y_t))$.

2.5 Training

We train our model by minimizing the weighted-sum of the two losses:

$$Loss = Loss_d + \gamma Loss_p \quad (15)$$

where γ is a hyper-parameter controlling the impact of $Loss_p$. Since minimizing cross-entropy loss does not always produce the best results due to the exposure bias (Ranzato et al., 2015), we further adopt the self-critical sequence training (SCST) algorithm (Rennie et al., 2017), which is a reinforcement learning process with the reward obtained by the current model.

Specifically, we produce two separate output sequences at each training iteration: (1) the sampling output y^s , which is obtained by sampling from the output distribution $P(y_t)$ at each decoding time step, and (2) the baseline output \hat{y} , which is obtained by maximizing the output distribution with a greedy search. We define $r(y)$ as the reward function, which is computed by comparing an output sequence y with the ground truth sequence using the evaluation metric of our choice. The SCST loss is given by

$$Loss_{rl} = -(r(y^s) - r(\hat{y})) \sum_{t=1}^T \log(P(y_t)) \quad (16)$$

Thus, minimizing $Loss_{rl}$ is equivalent to maximizing the conditional likelihood of sampled sequence y^s if it obtains a higher reward than the baseline \hat{y} , which improves the reward expectation of our model.

3 Experimental Setup

3.1 Datasets

We perform experiments on three public multi-turn task-oriented dialog datasets: In-Car Assistant (Eric and Manning, 2017), CamRest (Wen et al., 2016) and Multi-WOZ 2.1 (Qin et al., 2020).

In-Car Assistant dataset consists of 3,031 multi-turn dialogs in three distinct domains: schedule (Sch.), weather (Wea.), and navigation (Nav.). This dataset has an average of 2.6 turns, and the KB information is complicated, which has an average of 62.3 triples for every dialog. Following the data processing in Madotto et al. (2018), we obtain 2,425/302/304 dialogs for training/validation/test respectively.

CamRest dataset consists of 676 conversations in the restaurant reservation domain with 5.1 turns on average. It has an average of 22.5 KB triples for every dialog. Following the data processing in Raghu et al. (2019), we divide the dataset into training/validation/test sets with 406/135/135 dialogs respectively.

Multi-WOZ 2.1 dataset extends the Multi-WOZ (Budzianowski et al., 2018) by equipping the corresponding KB to every dialog, making it directly applicable for end-to-end generation. Following the data processing in Qin et al. (2020), we obtain 1,839/117/141 dialogs for training/validation/test respectively. It contains three distinct domains, including restaurant (Res.), attraction (Att.), and hotel (Hot.). These dialogs have 5.6 turns on average, with an average of 54.4 KB triples for every dialog.

3.2 Implementation Details

Our model is trained in an end-to-end manner using Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 5×10^{-4} . The shared size of embedding and the hidden units of GRU are sampled from $\{128, 256\}$. Both the number of rounds R and the number of hops K are set to 3. The dropout rate is set between $[0.1, 0.4]$ and the hyper-parameter γ in the loss function is set to 1. The hyper-parameters are tuned with grid-search over the validation set using BLEU score as metric. We select the model with best BLEU score as an initialization for SCST training, and use the weighted sum of BLEU and entity F1 score as our reward metric. During the decoding stage, we use beam-search strategy with the beam size sampling from $\{1, 2, 4\}$.

3.3 Baselines

We compare our model with several existing end-to-end task-oriented dialog systems: (1) **Seq2Seq+Attn** that employs standard seq2seq with and without attention over the input context (Luong et al., 2015); (2) **Ptr-Unk** that employs a seq2seq model with a copy mechanism to copy unknown words during generation (Gulcehre et al., 2016); (3) **Mem2Seq** that employs a memory network based approach with multi-hop attention for attending over dialog history and KB triples (Madotto et al., 2018); (4) **BossNet** that employs a bag-of-sequences memory network for disentangling language model from KB incorporation in task-oriented dialogs (Raghu et al., 2019); (5) **MLM** that employs a multi-level memory network for modeling dialog context and KB results separately (Reddy et al., 2019); (6) **GLMP** that employs a memory network with a global memory pointer and a local memory pointer to strengthen the copy ability (Wu et al., 2019).

3.4 Evaluation Metrics

Following previous works (Madotto et al., 2018; Wu et al., 2019), we evaluate our model and other baselines on two automatic evaluation metrics: **BLEU** (Papineni et al., 2002) and **Entity F1**. BLEU calculates n -gram overlaps between the generated response and the gold response. Entity F1 is computed by micro-averaging the precision and recall over KB entities in the entire set of system responses, which evaluates the performance of the model to generate relevant entities to achieve specific tasks from the provided KBs. It is noteworthy that entity F1 indicates the task-completion ability of the model, since KB entities are the key towards the dialog task.

Model	In-Car Assistant					Multi-WOZ 2.1					CamRest	
	BLEU	F1	Sch. F1	Wea. F1	Nav. F1	BLEU	F1	Res. F1	Att. F1	Hot. F1	BLEU	F1
Seq2Seq	8.4	10.3	9.7	14.1	7.0	4.3	9.2	10.5	8.7	8.2	7.9	17.6
Seq2Seq+Attn	9.3	19.9	23.4	25.6	10.8	4.5	11.6	11.9	10.8	11.1	7.7	21.4
Ptr-Unk	8.3	22.7	26.9	26.7	14.9	4.8	17.4	19.6	16.6	15.5	5.1	16.4
Mem2Seq	12.6	33.4	49.3	32.8	20.0	6.6	21.6	22.4	22.0	21.0	13.5	33.6
BossNet	8.3	35.9	50.2	34.5	21.6	5.7	25.3	26.2	24.8	23.4	15.2	43.1
MLM	15.6	55.5	67.4	54.8	45.1	9.2	27.8	29.8	27.4	25.2	16.1	55.2
GLMP	14.8	59.9	69.6	62.5	53.0	6.9	32.4	38.4	24.4	28.1	16.7	52.3
DDMN (Ours)	17.7	55.6	65.0	58.7	47.2	12.4	31.4	30.6	32.9	30.6	19.3	58.9
DDMN+SCST (Ours)	15.8	60.7	69.3	64.7	53.2	11.5	34.2	38.5	34.1	31.1	18.7	59.1

Table 2: Evaluation results on different datasets.

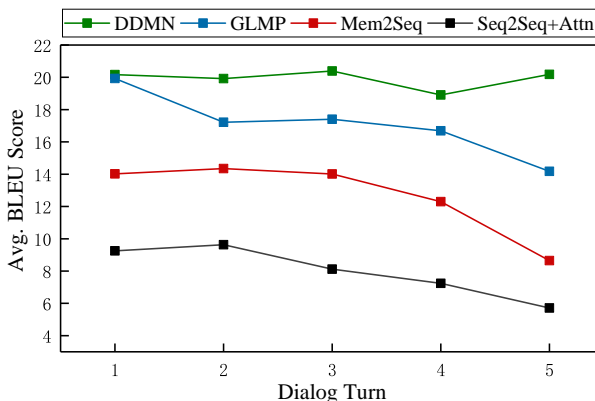


Figure 2: Comparison between DDMN and baselines on CamRest by varying the number of dialog turns.

4 Experimental Results

4.1 Quantitative Results

Automatic Evaluation Table 2 shows the evaluation results on different datasets, we can observe that our framework achieves the state-of-the-art performance in terms of BLEU and overall entity F1 on all datasets. On In-Car Assistant dataset, BossNet obtains better entity F1 than Mem2Seq but with a much lower BLEU score. By analyzing the responses generated by BossNet, we reveal that BossNet tends to copy necessary entity words from the KB but many entity words are out of order compared with the gold response. MLM achieves a much higher BLEU score than previous models, which is due to its separate memories for modeling dialog context and KB results. GLMP has achieved a strong improvement on BLEU and entity F1, which is mainly benefited from its global and local memory pointers to guide the KB attention and response generation. Our models perform even better than GLMP, which verifies the effectiveness of our models in generating natural and appropriate responses. We observe similar trend on Multi-WOZ 2.1 dataset. In particular, our model achieves significant higher BLEU score than other methods. It is because the conversations in Multi-WOZ 2.1 require long-turn interactions, which further shows the effectiveness of our framework in generating correct responses.

On CamRest dataset, our models substantially and consistently outperform the baseline methods by a noticeable margin. It is noteworthy that DDMN achieves significant improvement with a highest BLEU of 19.3 and promising entity F1 score of 58.9%, while DDMN with SCST obtains the best entity F1 score of 59.1% but has a lower BLEU score. This may be because that optimizing the combined discrete reward metrics in SCST does not guarantee an increase in quality of the output, since BLEU measures the n -gram overlap while entity F1 score captures the entity words without considering the order of words. Furthermore, Figure 2 shows the changes of average BLEU scores of DDMN and several baselines along with the increase of dialog turns on CamRest dataset. The BLEU scores of the baseline models decrease sharply as the dialog turns increase while DDMN achieves much more stable performance, suggesting that certain dialog and KB modeling strategy devised in DDMN to keep track of the dialog context, KB knowledge, and previous *inference* process is effective.

Model	Turn-level		Dialog-level	
	Appr.	Humanlike.	Goal.	Coherence
Mem2Seq	3.15	3.24	2.49	2.62
BossNet	3.32	3.78	3.25	3.04
GLMP	4.07	3.85	3.92	3.12
DDMN	4.12	4.13	4.17	3.65

Table 3: Human evaluation results at the level of turns and dialogs.

Restaurant	Area	Address	Food	Price	Phone	Postcode
ask	centre	12.bridge_street.city_centre	italian	cheap	01223_364917	cb21uf
da_vinci.pizzeria	north	20.milton_road.chesterton	italian	cheap	01223_351707	cb41jy
...

Role	Turn	Utterance
User	1	hello, i am looking for a cheap restaurant that serves eritrean food.
System	1	there is not a restaurant serving eritrean food listed. would you like something else?
Mem2Seq	1	ask is a cheap restaurant in the in the cheap price range.
GLMP	1	la_margherita is a cheap restaurant in the eritrean part of town.
DDMN	1	there are no eritrean restaurants in the centre of town. would you like to try something else?
User	2	yes, are there any restaurants that serve italian food?
System	2	there are five cheap restaurants that serve italian food. what area would you like to search?
Mem2Seq	2	ask is a cheap restaurant in the cheap price range and in the cheap price range.
GLMP	2	da_vinci.pizzeria serves italian food in the cheap part of town and the centre price range.
DDMN	2	there is an cheap italian restaurant in the north part of town. would you like the address and phone number?
User	3	i do not have a preference.
System	3	ask is a cheap italian restaurant located in the centre part of town.
Mem2Seq	3	ask is a cheap restaurant in the cheap price range and in the cheap price range. .
GLMP	3	ask is in the cheap part of town the centre part of town.
DDMN	3	there are two italian restaurants in the centre of town.
User	4	i need their address and phone number please.
System	4	the phone number for ask is 01223_364917 , and they are located at 12.bridge_street.city_centre .
Mem2Seq	4	their address is 15_magdalene_street.city_centre and their phone number is 01223_315232. .
GLMP	4	ask is located at 15_magdalene_street.city_centre and their phone number is 01223_315232.
DDMN	4	their address is 12.bridge_street.city_centre and their phone number is 01223_364917 .

Table 4: Case study of generated responses by DDMN and baselines on CamRest test set. The gold entities in each response are highlighted in bold.

Human Evaluation We randomly select 100 dialogs from the test data of different datasets for human evaluation. Following Wu et al. (2019), we adopt the **appropriateness** (*Appr.*) and **human-likeness** (*Humanlike.*) to judge the quality of the generated responses at the turn level. We also adopt the **goal completion** (*Goal.*) and **coherence** to judge the completion of overall dialog task and the overall fluency of the whole dialog at the dialog level. Three annotators are invited to independently assign the score scaled from 0 to 5 for each generated output. We report the average rating scores from all annotators and the results are shown in Table 3. The agreement ratio computed with Fless’ kappa (Fleiss, 1971) is 0.57, showing moderate agreement. As shown in Table 3, DDMN outperforms the baseline methods on both turn level and dialog level, which is consistent with the automatic evaluation. In particular, DDMN obtains significant higher goal completion score and coherence score than compared methods, demonstrating the effectiveness of DDMN in modeling multi-turn interactions in task-oriented dialog generation.

4.2 Case Study

As an intuitive way to show the performance of task-oriented dialog systems, Table 4 reports some responses generated by DDMN and baseline models. We observe that Mem2Seq tends to generate repeated or inappropriate responses. For example, the responses in the first three turns generated by MemSeq are very similar in both content and sentence structure. GLMP performs much better than Mem2Seq, while its performance deteriorate with the increase of dialog turns, e.g., GLMP fails to extract correct key entities in the third and fourth turns. Compared with GLMP, DDMN is able to generate more proper and natural responses even in the last few turns during the conversation. This verifies that DDMN is capable of memorizing the key information from previous turns.

4.3 Model Ablation

To investigate the effectiveness of each module proposed in our framework, we conduct ablation test from four aspects, the results are reported in Table 5. First, we remove the dialog state memory updating (w/o DSMU), resulting in significant performance degradation on both In-Car Assistant and CamRest datasets. This suggests that updating dialog context turn by turn is a necessary step to distill important information

Model	In-Car Assistant		CamRest	
	BLEU	F1	BLEU	F1
DDMN	17.7	55.6	19.3	58.9
w/o DSMU	15.2	48.2	16.1	49.0
w/o g_1	16.5	49.2	18.3	50.1
w/o g_2	16.3	45.1	17.8	48.1
w/o Ptr_{kb}	17.1	50.3	18.8	52.2

Table 5: Ablation results of DDMN on two datasets.

for response generation. Second, we remove the two gates g_1 and g_2 in decoder separately, where g_1 controls a word generated from the vocabulary or copied from the dialog history memory, and g_2 controls whether a word should be copied from the KB memory. The results show that g_2 contributes more to the performance of DDMN than g_1 , since entity words can not be copied efficiently without g_2 . Finally, we remove the KB memory pointer (w/o Ptr_{kb}) during training, the performance drops slightly over both BLEU and entity F1 score.

4.4 Error Analysis

To better understand the limitations of the proposed model, we carry out an analysis of the errors made by DDMN. Specifically, we randomly select 100 responses generated by DDMN that achieve low human evaluation scores in the test set of In-Car Assistant. We reveal several reasons of the low evaluation scores, which can be divided into four categories. (1) KB entries in generated responses are incorrect (33%), which occurs especially when the given KB triples are large, and it is difficult for the model to attend accurately over the KB memory. (2) Generated responses incorrectly achieve user goals (30%) since our model can not capture user intent well sometimes. (3) The sentence structure of generated responses are not appropriate (21%), which occurs when the model directly returns KB entries, even if more information should be asked. (4) Miscellaneous errors (16%), e.g., the generated responses are grammatically incorrect or conflict with the user input.

5 Related Work

End-to-end methods have shown promising results recently and attracted increasing attention since they are easily adapted to a new domain. Some approaches view the dialog history as a sequence using recurrent neural networks (Eric and Manning, 2017; Gulcehre et al., 2016), which also force the KB triples with the same pattern and make it hard to perform reasoning over them. To better handle KB triples in task-oriented dialogs, the memory network based architecture (Bordes et al., 2017) and its variants (Wu et al., 2017; Wu et al., 2018) have been proposed and shown promising results. Mem2Seq (Madotto et al., 2018) and GLMP (Wu et al., 2019) further augmented memory based methods by incorporating copy mechanism (Gulcehre et al., 2016), which enable the models copy words from past dialog utterances or from KB. These methods use a shared memory for the KB triples and the dialog utterances, making it difficult to reason over the memory and distinguish between the two forms of data.

Recently, there have been several works employing separate memories for modeling the dialog context and KB triples (Raghu et al., 2019; Reddy et al., 2019; Chen et al., 2019). For example, BossNet (Raghu et al., 2019) implicitly disentangled the language model from knowledge incorporation and thus enhance the ability of copying unseen KB entries. Multi-level memory model (Reddy et al., 2019) represented the KB results with a multi-level memory instead of the form of triples. WMM2Seq (Chen et al., 2019) adopted a working memory to interact with a dialog context memory and a KB memory. Nevertheless, existing methods still ignore the flow of history information during conversations, making it struggle to perform well in long-turn interactions. Different from the aforementioned methods, we propose a dialog memory manager and a KB memory manager to dynamically track the dialog context and KB triples, respectively.

6 Conclusion

In this paper, we propose a novel Dual Dynamic Memory Network (DDMN) with a dialog memory manager and a KB memory manager for multi-turn end-to-end task-oriented dialog systems. DDMN

encodes dialog context turn by turn and the dialog memory manager dynamically tracks the dialog history. The KB memory manager shares the KB information throughout the whole conversation with a KB memory pointer to softly distill relevant KB entries at each turn. In addition, we leverage self-critical sequence training to boost the performance of DDMN. Extensive experiments on three public dialog datasets demonstrate the superior performance of our model in both automatic and human evaluation.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC) (No. 61272200 and No. 61906185), Science and Technology Planning Project of Guangdong Province (No. 2017B030306016), Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515011705), the Youth Innovation Promotion Association of CAS.

References

- Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Xiuyi Chen, Jiaming Xu, and Bo Xu. 2019. A working memory model for task-oriented dialog response generation. In *ACL*, pages 2687–2693.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Mihail Eric 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.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *ACL*, pages 140–149.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421.
- Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*, pages 1468–1478.
- Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2174–2185.
- Fandong Meng, Zhaopeng Tu, Yong Cheng, Haiyang Wu, Junjie Zhai, Yuekui Yang, and Di Wang. 2018. Neural machine translation with key-value memory-augmented attention. In *IJCAI*, pages 2574–2580.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.
- Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. *arXiv preprint arXiv:2004.11019*.
- Dinesh Raghu, Nikhil Gupta, et al. 2019. Disentangling language and knowledge in task-oriented dialogs. In *NAACL-HLT*, pages 1239–1255.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

- Revanth Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. Multi-level memory for task oriented dialogs. In *NAACL-HLT*, pages 3744–3754.
- Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *EMNLP*, pages 2153–2162.
- Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.
- Chien-Sheng Wu, Andrea Madotto, Genta Winata, and Pascale Fung. 2017. End-to-end recurrent entity network for entity-value independent goal-oriented dialog learning. In *Dialog System Technology Challenges Workshop, DSTC6*.
- Chien-Sheng Wu, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2018. End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6154–6158.
- Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. In *ICLR*.
- 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.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102*.
- Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 27–36.