GraphMemDialog: Optimizing End-to-End Task-Oriented Dialog Systems Using Graph Memory Networks

Jie Wu[1], **Ian G. Harris**[1], **Hongzhi Zhao**[2]
[1]University of California, Irvine
[2]School of Computer and Information Technology, Beijing Jiao Tong University
wuj8@uci.edu, harris@ics.uci.edu, hzzhao@bjtu.edu.cn

Abstract

Effectively integrating knowledge into end-to-end task-oriented dialog systems remains a challenge. It typically requires incorporation of an external knowledge base (KB) and capture of the intrinsic semantics of the dialog history. Recent research shows promising results by using Sequence-to-Sequence models, Memory Networks, and even Graph Convolutional Networks. However, current state-of-the-art models are less effective at integrating dialog history and KB into task-oriented dialog systems in the following ways: **1**. The KB representation is not fully context-aware. The dynamic interaction between the dialog history and KB is seldom explored. **2**. Both the sequential and structural information in the dialog history can contribute to capturing the dialog semantics, but they are not studied concurrently. In this paper, we propose a novel Graph Memory Network (GMN) based Seq2Seq model, GraphMemDialog, to effectively learn the inherent structural information hidden in dialog history, and to model the dynamic interaction between dialog history and KBs. We adopt a modified graph attention network to learn the rich structural representation of the dialog history, whereas the context-aware representation of KB entities are learnt by our novel GMN. To fully exploit this dynamic interaction, we design a learnable memory controller coupled with external KB entity memories to recurrently incorporate dialog history context into KB entities through a multi-hop reasoning mechanism. Experiments on three public datasets show that our GraphMemDialog model achieves state-of-the-art performance and outperforms strong baselines by a large margin, especially on datatests with more complicated KB information.

1 Introduction

Task-oriented dialogue systems (TDSs), in contrast with chichat systems, help users complete a specific task with natural language, for example, inquiring about weather, reserving restaurants, or booking flights. In one specific domain, TDS takes dialog utterances and a knowledge base (KB) as input and produces responses by understanding dialog history, retrieving the most related KB entities, and generating readable sentences. Traditionally, these dialog systems have been built as a pipeline, with modules including spoken language understanding (SLU), dialog state tracking, action selection and language generation (Young et al. 2013). However, pipelined dialog systems usually suffer from the credit assignment problem (Yang, Zhang, and Erfani 2020) and easily lead to error propagation. Furthermore, they are not flexible enough to be adapted to new domains.

Recently, in order to address these issues, end-to-end TDSs (Serban et al. 2016; Wen et al. 2017) with sequence-to-sequence models (Seq2Seq) have attracted attention due to their great flexibility and good quality. The core idea of a Seq2Seq model is to leverage an encoder to directly map the dialog history and KB to a vector representation, which is then fed into a decoder to generate a response word by word. Later, memory networks (MemNN) are used to effectively incorporate KB information into the Seq2seq model (Madotto, Wu, and Fung 2018; Zhong, Xiong, and Socher 2018). Despite achieving promising results, these models are inherently weak at representing temporal dependencies between memories, so they ignore the association between KB entities. Subsequently, Banerjee and Khapra (2019) first proposed to use Graph Convolutional Networks (GCNs) to capture the rich structural information hidden in the dialog history and the KB, and achieved big improvement compared with Seq2Seq models with attention. Yang, Zhang, and Erfani (2020) employed a new recurrent cell architecture to allow representation learning on graphs. However, these models still suffer from two major issues: **1) Learning of context-aware KB representation**. Although GCNs show promising results at capturing graph structural information inherent in the KB, current state-of-the-art models still fail to fuse meaningful dialog context semantics into the KB representation. Our study shows that making it fully context-aware can significantly reduce the response errors especially on datasets with more complicated KB information. **2) Modeling sequential and structural dialog context concurrently**. Both type of information can be jointly beneficial to response generation by capturing distinct aspects of semantic information. Unfortunately, they have never been combined to promote performance in parallel for TDSs.

In this paper, we propose a novel Graph Memory Network (GMN) based end-to-end task-oriented dialog model, GraphMemDialog. The proposed model learns fully context-aware KB representations integrated with both KB graph structural information and dialog history semantics. Multi-hop reasoning further enables a deeper level of semantic modeling of KB entities. A multi-head graph attention network (GAT) is used to learn the intrinsic structural information in the dialog history, together with traditional RNNs, to jointly guide the response decoder to reduce response generation errors.

Our contributions are summarized as follows:

- We propose a novel GMN based end-to-end model to effectively incorporate external knowledge bases into TDSs by enabling fully context-aware KB entity representations. We design a learnable memory controller coupled with external KB entity memories to recurrently incorporate the dialog history context into KB entities through a multi-hop reasoning mechanism.
- We introduce a multi-head GAT to learn the intrinsic structural information in the dialog history to jointly guide the response decoder with the cooperation of traditional RNNs.
- Experiments on three benchmark datasets (i.e., CamRest, In-Car Assistant, MultiWOZ 2.1) demonstrate that our GraphMemDialog outperforms the state-of-the-art models especially on In-Car Assistant and MultiWOZ 2.1 datasets with more complicated KB information.

2 Related Work

Initially task-oriented dialog systems were implemented by using pipelined approaches designing each essential module individually, including natural language understanding (Chen et al. 2016), dialog state tracking (Zhong, Xiong, and Socher 2018; Wu et al. 2019), policy learning (Peng et al. 2018), and natural language generation (Chen et al. 2019; Huang et al. 2020). Later, in order to reduce human effort and scale up to new domains, fully data-driven end-to-end models were found to be promising to build domain-agnostic dialog systems based on recurrent neural networks (Zhao et al. 2017; Lei et al. 2018). Furthermore, MemNNs (Sukhbaatar et al. 2015) were employed to effectively incorporate dialog history context and knowledge bases (Madotto, Wu, and Fung 2018; Wu, Socher, and Xiong 2019). Multi-hop mechanisms further enabled MemNN to perform knowledge reasoning to select the most relevant entities for generating a dialog response. Eric and Manning (2017a) proposed the use of key-value memory networks to integrate a knowledge base by using a key memory to represent the subject and relation and a value memory to learn the object. Chen, Xu, and Xu (2019) introduced a working memory model to interact with two separate memories for dialog history and KB tuples. Gangi Reddy et al. (2019) proposed a multilevel memory architecture to capture the hierarchical properties of KBs. Although memory network based models show promising results, MemNNs suffer inherently from being weak at representing temporal dependencies between memories due to ignoring the utterance order. On the other hand, pointer memories with copy mechanism proposed by Madotto, Wu, and Fung (2018) can effectively integrate KBs into dialog responses. Wu, Socher, and Xiong (2019) incorporated a global pointer mechanism and achieved improved performance. Unfortunately, the rich structural information inherently in dialog history and the KB as defined by entity-entity relations is not considered by previous work.

Subsequently, Graph Convolutional Networks (GCNs) have emerged as state-of-the-art methods for modeling knowledge graphs where entities are treated as nodes and their relations are edges. Specifically, Banerjee and Khapra (2019) proposed to use GCNs to model the word dependencies associated with utterances, and entity relations in a knowledge base. Yang, Zhang, and Erfani (2020) proposed a new recurrent cell architecture to learn the dependency graph of the dialog history. Although those methods have modeled the structural information of the dialog history and KBs, they have ignored the strong correlation between them. Using dialog history as context should have a large impact on KB entity representations. He et al. (2020) modeled the association between the dialog history and KBs by using a Flow operation (Huang, Choi, and tau Yih 2019) and Relational Graph Convolutional Networks (RGCN) (Schlichtkrull et al. 2017) to learn the KB entity representation. However, our work differs from previous work significantly by proposing a novel GMN framework to learn context-aware KB entity representations.

Graph Memory Networks (GMNs) extend an end-to-end MemNN to have a structured dynamic memory organized as a graph of memory cells. Pham, Tran, and Venkatesh (2018) proposed this new structure to perform molecular activity prediction. Lu et al. (2020) employed a GMN framework to accomplish one-shot and zero-shot video object segmentation tasks. Khasahmadi et al. (2020) introduced a new memory layer for graph neural networks which learned hierarchical graph representations.

However, none of these GMN approaches models the knowledge base in a task-oriented dialog system. In this paper, inspired by Pham, Tran, and Venkatesh (2018) and Lu et al. (2020), we propose a novel GMN framework to effectively learn context-aware KB representations. To the best of our knowledge, our work is the first one that uses GMNs to incorporate context-aware knowledge graph structure into an end-to-end model for task-oriented dialog systems.

3 Proposed Model

Given a dialog between a user (**U**) and a system (**S**), t-turn dialog utterances are represented as $(U_1, S_1), (U_2, S_2), ..., (U_t, S_t)$. Dialogs are associated with knowledge triples in the format of ($subject$, $relation$, $object$) denoted as (h, r, t). The structural information of KB triples (h, r, t) is modeled as a knowledge graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E})$ with $h, t \in \mathcal{V}$ and $r \in \mathcal{E}$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of all entities and relations in $\mathcal{G}_k$, respectively. At the i^{th} dialog turn, our system input is dialog history $(U_1, S_1, ..., U_{i-1}, S_{i-1}, U_i)$ and the associated knowledge graph $\mathcal{G}_k$. The system output is the generation of the next system response S_i word by word.

3.1 Model Overview

We propose a novel GMN framework to combine the merits of MemNNs and GCNs to effectively learn context-aware KB representations. Our proposed model is composed of

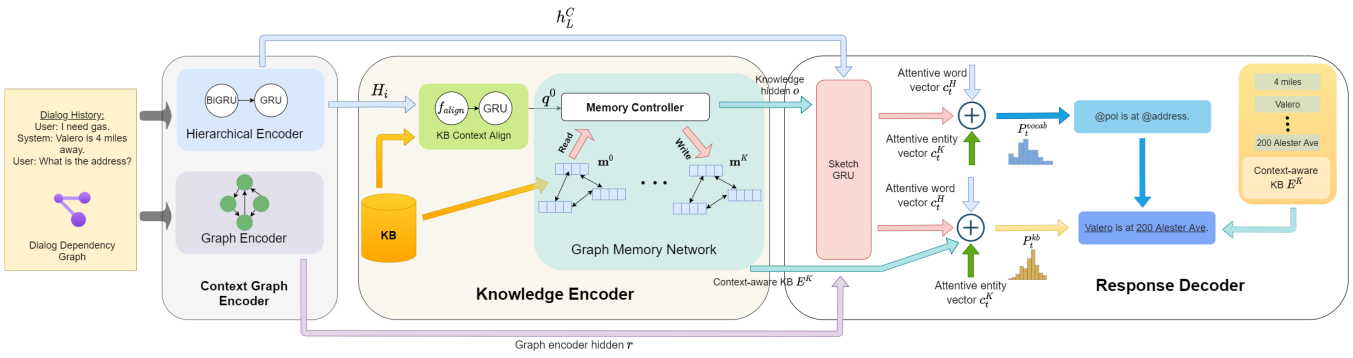

Figure 1: Framework of the proposed model

three major components: a context graph encoder, a knowledge encoder based on a GMN with multi-hopping reasoning, and a response decoder, as illustrated in Figure 1. The Context Graph Encoder learns a fixed-length vector to represent the dialog history both sequentially and structurally. We propose a graph encoder to encode the dialog history structural information, which is the dependency parsing graph of the sentences in the dialog history $\mathcal{G}_d$. Next the Knowledge Encoder encodes context-aware KB entity information by incorporating graph structure information and the dialog context semantics through our GMN and its multi-hop reasoning mechanism. Finally the Response Decoder generates the system response token-by-token, either by querying the knowledge graph, or by generating tokens from vocabularies under the constraint of the dialog and KB context. In the following sections we detail each component thoroughly.

3.2 Context Graph Encoder

Given that the dialog history has L utterances with each one containing T_i words, our context graph encoder encodes dialog context both sequentially and structurally via a hierarchical attention network (Yang et al. 2016) and a modified GAT (Veličković et al. 2018). This approach is different than previous work in which dialog history is encoded either sequentially or structurally. We observe that the combination of those two inputs is beneficial to effectively capture the semantic context, especially for multi-turn dialog systems.

Hierarchical Attention Encoder We employ a hierarchical attention network to sequentially encode the dialog history. A bidirectional RNN is applied to learn representations of each word w_{it} with $t \in [1, T_i]$ in the i^{th} utterance by reading the input utterance forward and backward to produce context sensitive hidden states.We use a BiGRU (Chung et al. 2014) to encode the dialog context into hidden states:

$$H_i = h_{i1}, ...h_{iT_i} = \textbf{BiGRU}(\phi^{emb}(w_{it}), h_{i(t-1)}) \tag{1}$$

where $\phi^{emb}(w_{it})$ is the embedding of the word w_{it}.

Then, we use self-attention mechanism to capture the contextual information for each token in order to get an interpretable utterance semantic representation as below:

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{2}$$

$$a_{it} = \frac{\exp(u_{it}^\top u_w)}{\sum_t \exp(u_{it}^\top u_w)} \tag{3}$$

$$\hat{s}_i = \sum_t a_{it} h_{it} \tag{4}$$

where W_w, b_w, u_w are trainable parameters of the model.

Finally, we use a GRU to encode the utterance vector $\hat{s}_i$:

$$h_i^C = \textbf{GRU}(\hat{s}_i), i \in [1, L] \tag{5}$$

Graph Encoder In order to enable learning various relationships of words such as dependency relations, we first use the off-the-shelf parsing tool called Spacy[1] to extract dependency relation graph $\mathcal{G}_d$ among words in dialog history. To model word nodes and relations jointly, we employ a modified variant of graph attention network (GAT) (Veličković et al. 2018), which is enhanced to model multi-relational edges. In this way, modified GAT learns attention not only from the neighboring nodes, but also from edge features. This is important because dialog data contains rich edge information.

We first apply another BiGRU to process all the concatenated words in dialog history to get their contextual representation, which are then fed in our modified GAT:

$$h_t^G = \textbf{BiGRU}(\phi^{emb}(w_t), h_{t-1}^G) \tag{6}$$

In the l-th layer of the modified GAT, the attention score between two neighboring words is obtained as follows:

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(W[W_a h_i^{G^{(l)}} \| W_a h_j^{G^{(l)}} \| W_e {h_{i\to j}^E}^{(l)}]\right)\right)}{\sum\limits_{k\in\mathcal{N}_i} \exp\left(\sigma\left(W[W_a h_i^{G^{(l)}} \| W_a h_k^{G^{(l)}} \| W_e {h_{i\to k}^E}^{(l)}]\right)\right)} \tag{7}$$

where ${h_{i\to j}^E}^{(l)}$ denotes the representation of the edge connecting word node i to its neighboring word node j. W_a and W_e are trainable word node and edge weights, whereas W is a single-layer feed-forward network parameter that computes the attention score. σ is the LeakyReLU activation

[1]https://spacy.io/

function. Word hidden states are obtained by multi-head attention mechanism via averaging:

$$\widetilde{h}_t^G = \sigma\left(\frac{1}{N}\sum_{n=1}^{N}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^n W_n h_j^G\right) \tag{8}$$

where N is the number of heads, and W_n is the corresponding input linear transformation's weight matrix.

Finally we apply the same procedure on $\widetilde{h}_t^G$ as shown in eqs. (2) to (4) to obtain our final integrated graph context information:

$$r = \text{self-attention}\left(\widetilde{h}_t^G\right) \tag{9}$$

Contrary to BiGRU only capturing sequential associations between words, dependency relationship is supposed to learn the mutual interaction between head words and dependent words. Those two types of relationship should complement each other.

3.3 Knowledge Encoder

The knowledge encoder obtains a context-aware representation of each entity in the knowledge graph. We propose a novel graph memory network framework by extending a graph with multi-hop reasoning to model knowledge entities, through which entity dependencies can be well modeled and fused with dialog history context.

Context Alignment In order to make our knowledge entity context-aware, we first align the embedding of entity $e_j, j \in [1, E]$ with that of word w_{it} in the i^{th} utterance as shown in He et al. (2020), where E is the number of entities in the KB:

$$f_{align}^i(e_j) = \sum_t \alpha_{jit}\phi^{emb}(w_{it}) \tag{10}$$

$$\alpha_{jit} = \frac{\exp\left(u_{jit}^\top u_e\right)}{\sum_t \exp\left(u_{jit}^\top u_e\right)} \tag{11}$$

$$u_{jit} = \tanh\left(W_e[\phi^{emb}(e_j)\|\phi^{emb}(w_{it})] + b_e\right) \tag{12}$$

where W_e, b_e, and u_e are trainable parameters of the model and $\|$ denotes the concatenation. Then we pass the entity sequence to a GRU as follows for entity e_j:

$$f_{ji} = \text{GRU}\left([\phi^{emb}(e_j)\|f_{align}^i(e_j)]\right), i \in [1, L] \tag{13}$$

After the above processing, each entity has L representations corresponding to L utterances, so $F = \{f_{ji}\} \in \mathbb{R}^{L\times E\times d_e}$, where d_e is the entity embedding dimension.

Furthermore, to have a deeper integration of the dialog history, we perform another level of knowledge entity alignment with our sequential encoder BiGRU hidden state h_{it} as shown in Equation (1), because it captures the temporal dependency between words.

$$h_{ji}^A = \text{GRU}\left([f_{ji}\|f_{align}(f_{ji}, h_{it})]\right), j \in [1, E] \tag{14}$$

We take the entity representations under the L^{th} utterance from the output of the second alignment process as our initial context-aligned entity representations, namely:

$$E^A = \{h_{1L}^A, ...H_{EL}^A\} \in \mathbb{R}^{E\times d_e} \tag{15}$$

Graph Memory Network Our GMN is composed of an external graph memory and learnable controllers for memory reading and writing. The memory graph structure with multi-hop mechanism enables strong reasoning ability on knowledge entities. The controllers interact with memories using read and write operations to carry long-term context information and to encode new knowledge via slow updates of the weights. Through iterations, our GMN learns a general strategy to represent knowledge entities under a specific dialog context, making them quite context-aware.

We incorporate the semantics contained in the historical context into the KB entity memory slots. The memory is organized as a fully connected graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E})$, where node $m_i \in \mathcal{V}$ denotes i^{th} memory cell, which learns to represent entity e_i, and edge $\bar{e}_{ij} = (m_i, m_j) \in \mathcal{E}$ indicates the relation between entity e_i and e_j. The graph memory cells are initialized by the knowledge entity embedding $\{e_1, ..., e_E\}$. Subsequently these memory cells are augmented to capture the dialog history via controller writing.

Graph Memory Reading In order to effectively integrate context history into knowledge entities, we take context alignment output E^A as our initial state h_0^M to our GMN. A learnable read controller at each iteration step $k \in 1, ..., K$ interacts with graph memory by reading the content. m^k is a sum of all memory cells, weighted by the probability w_i^k. Formally (Lu et al. 2020):

$$b_i^k = \frac{h_{k-1}^M \cdot m_i^{k-1}}{\|h_{k-1}^M\|\|m_i^{k-1}\|} \tag{16}$$

$$w_i^k = \frac{\exp(b_i^k)}{\sum_j \exp(b_j^k)} \tag{17}$$

$$m^k = \sum_i w_i^k m_i^{k-1} \tag{18}$$

Once reading memory content, the read controller updates its state as follows:

$$\begin{aligned}\widetilde{h}_k^M &= W_r^h h_{k-1}^M + U_r^h m^k \\ a_r^k &= \sigma(W_r^a h_{k-1}^M + U_r^a m^k) \\ h_k^M &= a_r^k \widetilde{h}_k^M + (1 - a_r^k) h_{k-1}^M\end{aligned} \tag{19}$$

where W_r^h, U_r^h, W_r^a, U_r^a are trainable parameters of the model. The udpate gate a_r^k controls how much previous hidden state h_{k-1}^M to be kept. In this way, the hidden state of the controller encodes both the KB entity memory and dialog history representations, hence context-aware.

Graph Memory Updating After we obtain a new query h_k^M, we need to update the graph memory with the new query input. At each step k, each memory cell is augmented by a learnable write controller, which is a function of previous memory state m_i^{k-1}, current query state h_k^M, and the states from all of its neighboring cells, namely:

$$m_i^k = f\left(m_i^{k-1}, h_k^M, \left(m_j^{k-1}\right)_{j\in\mathcal{N}(i)}\right) \tag{20}$$

where $\mathcal{N}_i$ is the neighbors of the node m_i. Following Pham, Tran, and Venkatesh (2018), we calculate the summarized information c_i^k from neighboring entities as follows:

$$c_i^k = \sum_{j \in \mathcal{N}(i)} p_{i,j}^k \left[m_j^{k-1} \| \bar{e}_{i,j}^k \right] \tag{21}$$

where $\bar{e}_{i,j}^k$ is the relation feature vector between entities and $p_{i,j}^k$ is the weight of m_j, which indicates how important the node m_j towards m_i. $p_{i,j}^k$ can be learned, similarly to memory cell probabilities in the attentive reading as in Equations (16) and (17).

After aggregating the information from neighbors, the memory write controller updates the state of m_i as:

$$\begin{aligned} \widetilde{m}_i^k &= W_u^m h_k^M + U_u^m m_i^{k-1} + V_u^m c_i^k \\ a_u^k &= \sigma(W_u^a h_k^M + U_u^a m_i^{k-1} + V_u^a c_i^k) \\ m_i^k &= a_u^k \widetilde{m}_i^k + (1 - a_u^k) m^{k-1} \end{aligned} \tag{22}$$

The graph memory updating allows each memory cell to embed both the neighbor information and the dialog context information into its representation, making it fully context-aware. Moreover, by iteratively reasoning over the graph structure, each memory cell encodes the new query information and yields progressively improved representations. Those salient properties make our GMN overcome the shortness of traditional memory networks and also dwarf popular GCNs because of GMN's multi-hop reasoning.

Final GMN Outputs After K steps of iteration, we concatenate the output of context aligned KB entities and the final memory state m^K as our final knowledge entity representations:

$$E^K = \left[E^A \| m^K \right] \tag{23}$$

Also the first hidden state of h_K^M, that is $\mathbf{o} = h_K^M[0]$, is considered as our knowledge encoder hidden state to carry over KB context to the response decoder.

3.4 Response Decoder

The response decoder is conditioned on dialog sequential and structural context representation, and context-aware entity representation. Inspired by Wu, Socher, and Xiong (2019), we use a sketch GRU to generate a sketch response that excludes slot values but includes sketch tags, which are all possible slot types starting with a special token, for instance, @distance. The sketch GRU learns to generate a dynamic dialogue action template. For example, instead of generating "Stanford shopping center is 2 miles away at 773 alger dr", it produces "@poi is @distance away at @address". At each decoding timestep, if a sketch tag is generated, we select an appropriate entity as the output word by querying entity representation. Otherwise, the output word is the word generated by the sketch GRU. For example, if "@poi" tag is generated, the words "Stanford shopping center" is selected from our KB entities to replace this tag as part of our final response.

The initial hidden state of the decoder h_0^D consists of context graph encoder hidden and knowledge encoder hidden output, which further constrains the decoding process under the current dialog context.

$$h_0^D = \sigma \left(W_d \left[h_L^C \| o \| r \right] \right) \tag{24}$$

where W_d is the trainable parameter. At each decoding time step t, the GRU takes the previously generated s_{t-1} and the previous hidden state h_{t-1}^D as the input and generates a new hidden state h_t^D as follows:

$$h_t^D = \text{GRU} \left(\phi^{emb}(s_{t-1}), h_{t-1}^D \right) \tag{25}$$

In order to handle long-term dependency, we again use an attention mechanism to dynamically determine the importance of each word in the dialog history and each entity in the KB. At each time step t, the decoder generates an attentive entity vector based on context-aware entity representation E^K as follows:

$$\begin{aligned} \alpha_{ti} &= \frac{\exp\left({h_t^D}^\top \mathrm{W}_k e_i^K \right)}{\sum_j \exp\left({h_t^D}^\top \mathrm{W}_k e_j^K \right)} \\ c_t^K &= \sum_{i=1}^{E} \alpha_{ti} e_i^K \end{aligned} \tag{26}$$

Similarly, we generate an attentive word vector c_t^H based on H_L specified in Equation (1). Finally, the decoder generates two distributions, that is, a vocabulary distribution P_t^{vocab}, and a knowledge entity distribution P_t^{kb} to either select a vocabulary word or an entity word from the KB:

$$P_t^{vocab} = \text{Softmax} \left(W_v \left[h_t^D \| c_t^K \| c_t^H \right] \right) \tag{27}$$

$$P_t^{kb} = \text{Softmax} \left({E^K}^\top W_{kb} \left[h_t^D \| c_t^K \| c_t^H \right] \right) \tag{28}$$

where W_v and W_{kb} are trainable parameters.

3.5 Joint Training

Following Wu, Socher, and Xiong (2019), we replace the slot values in the response S with sketch tags based on the provided entity table to create a sketch response $S^s = (s_1^s, ..., s_m^s)$. We use the standard negative log-likelihood loss to train the sketch GRU as:

$$\mathcal{L}_1 = \sum_{t=1}^{m} -\log(P_t^{vocab}(s_t^s)) \tag{29}$$

Similarly, we have another loss for our KB entities which eventually replace all the sketch tags to form a final response:

$$\mathcal{L}_2 = \sum_{t=1}^{m} -\log(P_t^{kb}(s_t^e)) \tag{30}$$

where $\{s_t^e\}$ represents the entity sequence. For the case when s_t is not an entity token, we train P_t^{kb} to produce a special token.

Finally the joint objective is formulated as the weighted-sum of these two loss functions using hyper-parameters α and β:

$$\mathcal{L}_\theta = \alpha \mathcal{L}_1 + \beta \mathcal{L}_2 \tag{31}$$

4 Experimental Setup

4.1 Datasets

To evaluate our proposed model, we conduct experiments on three widely used benchmark datasets: CamRest (Wen et al. 2016), In-Car Assistant (Eric and Manning 2017b), and Multi-WOZ 2.1 (Qin et al. 2020).

CamRest This dataset contains dialogs in the restaurant reservation domain, involving 676 multi-turn dialogs and having 5 turns on average per dialog (Wen et al. 2016). It also has an average of 22.5 KB triples for every dialog. We divide this dataset into training/validaton/test sets with 406/135/135 dialogs, respectively, as Raghu, Gupta, and Mausam (2019) did.

In-Car Assistant This dataset consists of 3,031 multi-turn dialogs in three distinct domains: weather (Wea.), navigation (Nav.), and schedule (Sch.) This dataset has an average of 2.6 turns. However the KB information is more compalicated than CamRest with an average of 62.3 triples for every dialog. Following Madotto, Wu, and Fung (2018), we divide the In-Car Assistant dataset into training/validaton/test sets with 2425/302/304 dialogs, respectively.

Multi-WOZ 2.1 This dataset contains three distinct domains: attraction (Att.), hotel (Hot.) and restaurant (Res.), with an average of 5.6 turns and 54.4 KB triples per dialog. Following how Qin et al. (2020) processed data, we have 1,839/117/141 dialogs for training/validation/test.

4.2 Training Details

We implement our model in Pytorch, which is trained on an NVIDIA GeForce RTX 2080 Ti. In our experiments, we set all the embedding dimension and hidden units to 200 and batch size to 8. The model is trained end-to-end using the Adam optimizer (Kingma and Ba 2014) and learning rate annealing starts from $1e^{-3}$ to $5 \times 1e^{-5}$. Embeddings are randomly initialized and updated during training. For all the datesets, the dropout ratio is set to 0.5, the number of our GAT's attention heads is set to 6, and the number of hops for our GMN is set to be 2.

4.3 Baselines

We compare our proposed GraphMemDialog model with several representative works: (1) **Seq2Seq+Attn** (Luong, Pham, and Manning 2015); (2) **Mem2Seq** (Madotto, Wu, and Fung 2018); (3) **GLMP** (Wu, Socher, and Xiong 2019); (4) **DDMN** (Wang et al. 2020); (5) **FG2Seq** (He et al. 2020); (6) **MCL** (Qin et al. 2021). When doing the comparison, we adopt reported results from those papers directly.

4.4 Automatic Evaluation Metrics

In order to have fair comparison with others' work, we adopt the two most popular evaluation metrics in dialogue studies (Zhong, Xiong, and Socher 2018; Madotto, Wu, and Fung 2018; Qin et al. 2021).

1. **Bilingual Evaluation Understudy (BLEU)** (Papineni et al. 2002). BLEU computes the n-gram overlap between the produced responses and gold ones.
2. **F1 Score (Entity F1)**. The entity F1 score is generally used to measure the system's capability of generating relevant entities to accomplish certain tasks by retrieving accurate entities from the provided KB. The entity F1 score is computed by micro-averaging the precision and recall over KB entities of the generated responses (Qin et al. 2021).

5 Experimental Results

5.1 Automatic Evaluation Results

Table 1 shows the experiment results of the proposed model on CamRest, In-Car Assistant, and Multi-WOZ 2.1 datasets. From the table, we can see that our model substantially outperforms all the baselines by a noticeable margin on both BLEU score and entity F1, demonstrating that our context-aware graph memory network can benefit dialog response generation effectively. On the single domain CamRest dataset, compared with the best prior work Fg2Seq He et al. (2020), we achieve performance improvement by 10% on BLEU, and almost 4% on entity F1, respectively. The performance jump on BLEU score signifies that our decoder's generation error has been greatly reduced, whereas the gain on entity F1 indicates that our model can retrieve entities from the external knowledge data more accurately than those baselines. This demonstrates that our GraphMemDialog model can not only improve the dialog history context modeling by capturing the graph structure in dialogs via graph attention networks, but also effectively model the interaction between the dialog history and KB entities, making them fully context-aware.

On both In-Car assistant and Multi-WOZ 2.1 datasets, our GraphMemDialog also outperforms all the other baselines by a large margin both in BLEU score and entity F1, which indicates that our model has a better generalization capability than baseline models. Our model outperforms Fg2Seq by 12% on BLEU score and 5.6% on entity F1 on In-Car Assistant, and by 10% on BLEU score and 12% on entity F1 on Multi-WOZ 2.1. The gain on entity F1 further demonstrates our GMN's great reasoning capability under different dialog history contexts, especially considering In-Car assistant and Multi-WOZ 2.1 have much more complicated KB information. Even though Fg2seq and MCL have already made a great advancement in performance, our GraphMemDialog still outperforms them by a large margin.

Table 2 reports some responses generated by GraphMemDialog and some baseline models. Compared with GLMP and FG2Seq, GraphMemDialog is more effective at carrying over dialog context to next turns and generating context-aware responses. For example, in the second turn, since the query is very short, GLMP and Fg2Seq tends to generate unrelated responses. GraphMemDialog shows strong capability to extract key entities, whereas GLMP fails to fill slot tag @weather_attribute it has produced. We attribute those merits mainly to our GMN's contributions.

5.2 Ablation Study

In this section, we explore how each component contributes to our full model. We conduct some ablation tests by re-

Model	CamRest		In-Car Assistant					Multi-WOZ 2.1				
	BLEU	**Ent.F1**	**BLEU**	**Ent.F1**	**Sch.F1**	**Wea.F1**	**Nav.F1**	**BLEU**	**Ent.F1**	**Res.F1**	**Att.F1**	**Hot.F1**
Seq2Seq+Attn	7.7	21.4	9.3	19.9	23.4	25.6	10.8	4.5	11.6	11.9	10.8	11.1
Mem2Seq	13.5	33.6	12.6	33.4	49.3	32.8	20.0	6.6	21.6	22.4	22.0	21.0
GLMP	16.7	50.6	14.8	60.0	69.6	**62.6**	53.0	6.9	32.4	38.4	24.4	28.1
DDMN	19.3	58.9	17.7	55.6	65.0	58.7	47.2	12.4	31.4	30.6	32.9	30.6
MCL	20.1	59.2	17.2	60.9	70.6	**62.6**	**59.0**	13.6	32.6	34.4	30.2	29.8
Fg2Seq	20.2	62.1	16.8	61.1	73.3	57.4	56.1	13.5	36.0	40.4	41.7	30.9
GraphMemDialog	**22.3**	**64.4**	**18.8**	**64.5**	**75.9**	62.3	56.3	**14.9**	**40.2**	**42.8**	**48.8**	**36.4**

Table 1: Performance evaluation results on CamRest, In-Car Assistant, and Multi-WOZ 2.1 datasets.

Knowledge Base	today is monday downtown chicago monday hail downtown chicago monday low 50f downtown chicago monday high 70f	
Role	**Turn**	**Utterance**
User	1	give me weather forecast for today.
System (Gold)	1	what city do you want the weather forecast for?
GLMP	1	what city do you want the weather for?
Fg2Seq	1	what city would you like to hear the forecast for?
GraphMemDialog	1	what city are you interested in?
User	2	downtown chicago, please.
System (Gold)	2	**today** in **downtown chicago** there should be **hail** with a high of **70f**.
GLMP	2	it will be **downtown chicago** in **today** monday.
Fg2Seq	2	what would you like to know about **today**?
GraphMemDialog	2	**today** in **downtown chicago** it will be **hail** today, and **hail** with a low of 50f and a high of **70f**.

Table 2: Responses generated by GraphMemDialog and some baseline models on In-Car Assistant dataset. The gold entities in each response are highlighted in bold.

Model	CamRest		In-Car Assistant		Multi-WOZ 2.1	
	BLEU	**Ent.F1**	**BLEU**	**Ent.F1**	**BLEU**	**Ent.F1**
GraphMemDialog	**22.3**	**64.4**	**18.8**	**64.5**	**14.9**	**40.2**
w/o GMN	20.8	56.2	16.7	52.0	14.1	31.2
w/o GAT	21.2	62.4	18.8	63.6	14.2	39.0
w/o Both	20.2	54.8	16.4	50.6	12.9	30.8

Table 3: Ablation results of GraphMemDialog on CamRest, In-Car Assistant, and Multi-WOZ 2.1 datasets.

moving GMN (w/o GMN), and modified GAT (w/o GAT). Table 3 shows the performance changes. Firstly, if we only remove GMN, which means no KB structural information and no iterative interaction between dialog hisotry and KB involved, the performance degrades dramatically, especially on entity F1. This further demonstrates that our GMN makes a major contribution to our performance improvement. This is due to the fact that GMN not only learns graph structure inherent in KB, but also models the interaction between the dialog history and KB effectively as fully context-aware, enhancing the possibility to retrieve the most relevant entities from the KB. Next, if we only remove modified GAT, it is noticeable that the performance is degraded, but not significantly. Also we can observe that GAT is more helpful improving BLEU score than lifting entity F1. We attribute this to the fact that GAT is mainly for capturing structural information in the dialog history which helps reduce decoder generation errors. Finally, if we remove both modules, it is not surprising that the performance drops dramatically. This verifies that our GraphMemDialog model makes a significant contribution to modeling context-aware external knowledge base, and capturing the structural information in dialog history.

5.3 Comparison with Conventional GCNs

Our proposed GMN shows improvement on modeling context-aware KB. Due to its multi-hop reasoning, we expect it to be superior to conventional GCNs. In order to verify this, we design some experiments to replace our GMN with two widely-used GCNs: Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al. 2017), and Composition-based Multi-relational Graph Convolutional Network (COMPGCN) (Vashishth et al. 2020). To be fair, we carefully choose GCNs which can encode relations as well, since our GMN does that. From Table 4, it is noticeable that our GMN outperforms COMPGCN and RGCN on both BLEU score and entity F1 on all three datasets in almost all cases. We attribute this to our GMN's multi-hop reasoning capability that effectively fuses dialog history context into KB entity representation, making it fully context-aware.

Model	CamRest		In-Car Assistant		Multi-WOZ 2.1	
	BLEU	**Ent.F1**	**BLEU**	**Ent.F1**	**BLEU**	**Ent.F1**
GraphMemDialog	**22.3**	**64.4**	**18.8**	**64.5**	**14.9**	40.2
COMPGCN	21.1	60.7	18.0	61.8	13.7	**40.3**
RGCN	21.3	60.3	18.1	63.4	14.1	36.3

Table 4: Performance comparison of GraphMemDialog with representative GCNs.

6 Conclusion

In this paper, we present a Graph Memory Network based end-to-end model for task-oriented dialog systems. GraphMemDialog models context-aware KB entities, and learns graph structure information hidden in dialog history and KBs. To fully fuse dialog context information into the KB, we design a learnable memory controller coupled with an external KB entity memory to recurrently incorporate the dialog history context into KB entities via a multi-hop reasoning mechanism. A modified GAT is employed to effectively capture graph structure information inherent in dialog history. Experiments on three public datasets show the effectiveness of our proposed model and achieve state-of-the-art results.

References

Banerjee, S.; and Khapra, M. M. 2019. Graph convolutional network with sequential attention for goal-oriented dialogue systems. *Transactions of the Association for Computational Linguistics*, 7: 485–500.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3696–3709. Florence, Italy: Association for Computational Linguistics.

Chen, X.; Xu, J.; and Xu, B. 2019. A working memory model for task-oriented dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2687–2693.

Chen, Y.-N. V.; Hakkani-Tür, D.; Tur, G.; Gao, J.; and Deng, L. 2016. End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*. ISCA.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555.

Eric, M.; and Manning, C. D. 2017a. Key-Value Retrieval Networks for Task-Oriented Dialogue. arXiv:1705.05414.

Eric, M.; and Manning, C. D. 2017b. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Gangi Reddy, R.; Contractor, D.; Raghu, D.; and Joshi, S. 2019. 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, Volume 1 (Long and Short Papers)*, 3744–3754. Minneapolis, Minnesota: Association for Computational Linguistics.

He, Z.; He, Y.; Wu, Q.; and Chen, J. 2020. Fg2seq: Effectively Encoding Knowledge for End-To-End Task-Oriented Dialog. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8029–8033.

Huang, H.-Y.; Choi, E.; and tau Yih, W. 2019. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. arXiv:1810.06683.

Huang, X.; Qi, J.; Sun, Y.; and Zhang, R. 2020. MALA: Cross-Domain Dialogue Generation with Action Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 7977–7984.

Khasahmadi, A. H.; Hassani, K.; Moradi, P.; Lee, L.; and Morris, Q. 2020. Memory-Based Graph Networks. arXiv:2002.09518.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1437–1447. Melbourne, Australia: Association for Computational Linguistics.

Lu, X.; Wang, W.; Danelljan, M.; Zhou, T.; Shen, J.; and Van Gool, L. 2020. Video object segmentation with episodic graph memory networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 661–679. Springer.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. arXiv:1508.04025.

Madotto, A.; Wu, C.-S.; and Fung, P. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. arXiv:1804.08217.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Peng, B.; Li, X.; Gao, J.; Liu, J.; and Wong, K.-F. 2018. Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2182–2192. Melbourne, Australia: Association for Computational Linguistics.

Pham, T.; Tran, T.; and Venkatesh, S. 2018. Graph memory networks for molecular activity prediction. In *2018 24th International Conference on Pattern Recognition (ICPR)*, 639–644. IEEE.

Qin, B.; Yang, M.; Bing, L.; Jiang, Q.; Li, C.; and Xu, R. 2021. Exploring Auxiliary Reasoning Tasks for Task-oriented Dialog Systems with Meta Cooperative Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15): 13701–13708.

Qin, L.; Xu, X.; Che, W.; Zhang, Y.; and Liu, T. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. *arXiv preprint arXiv:2004.11019*.

Raghu, D.; Gupta, N.; and Mausam. 2019. Disentangling Language and Knowledge in Task-Oriented Dialogs. arXiv:1805.01216.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2017. Modeling Relational Data with Graph Convolutional Networks. arXiv:1703.06103.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. arXiv:1507.04808.

Sukhbaatar, S.; Szlam, A.; Weston, J.; and Fergus, R. 2015. End-To-End Memory Networks. arXiv:1503.08895.

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. 2020. Composition-based Multi-Relational Graph Convolutional Networks. arXiv:1911.03082.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903.

Wang, J.; Liu, J.; Bi, W.; Liu, X.; He, K.; Xu, R.; and Yang, M. 2020. Dual Dynamic Memory Network for End-to-End Multi-turn Task-oriented Dialog Systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4100–4110. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Wen, T.-H.; Vandyke, D.; Mrksic, N.; Gasic, M.; Rojas-Barahona, L. M.; Su, P.-H.; Ultes, S.; and Young, S. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Wen, T.-H.; Vandyke, D.; Mrksic, N.; Gasic, M.; Rojas-Barahona, L. M.; Su, P.-H.; Ultes, S.; and Young, S. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. arXiv:1604.04562.

Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 808–819. Florence, Italy: Association for Computational Linguistics.

Wu, C.-S.; Socher, R.; and Xiong, C. 2019. Global-to-local Memory Pointer Networks for Task-Oriented Dialogue. arXiv:1901.04713.

Yang, S.; Zhang, R.; and Erfani, S. 2020. GraphDialog: Integrating Graph Knowledge into End-to-End Task-Oriented Dialogue Systems. arXiv:2010.01447.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489. San Diego, California: Association for Computational Linguistics.

Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5): 1160–1179.

Zhao, T.; Lu, A.; Lee, K.; and Eskenazi, M. 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*, 27–36. Saarbrücken, Germany: Association for Computational Linguistics.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1458–1467. Melbourne, Australia: Association for Computational Linguistics.