Graph Convolutional Network with Sequential Attention for Goal-Oriented Dialogue Systems

Suman Banerjee and **Mitesh M. Khapra**

Department of Computer Science and Engineering,
Robert Bosch Centre for Data Science and Artificial Intelligence (RBC-DSAI),
Indian Institute of Technology Madras, India
{suman, miteshk}@cse.iitm.ac.in

Abstract

Domain-specific goal-oriented dialogue systems typically require modeling three types of inputs, namely, (i) the knowledge-base associated with the domain, (ii) the history of the conversation, which is a sequence of utterances, and (iii) the current utterance for which the response needs to be generated. While modeling these inputs, current state-of-the-art models such as Mem2Seq typically ignore the rich structure inherent in the knowledge graph and the sentences in the conversation context. Inspired by the recent success of structure-aware Graph Convolutional Networks (GCNs) for various NLP tasks such as machine translation, semantic role labeling, and document dating, we propose a memory-augmented GCN for goal-oriented dialogues. Our model exploits (i) the entity relation graph in a knowledge-base and (ii) the dependency graph associated with an utterance to compute richer representations for words and entities. Further, we take cognizance of the fact that in certain situations, such as when the conversation is in a code-mixed language, dependency parsers may not be available. We show that in such situations we could use the global word co-occurrence graph to enrich the representations of utterances. We experiment with four datasets: (i) the modified DSTC2 dataset, (ii) recently released code-mixed versions of DSTC2 dataset in four languages, (iii) Wizard-of-Oz style CAM676 dataset, and (iv) Wizard-of-Oz style MultiWOZ dataset. On all four datasets our method outperforms existing methods, on a wide range of evaluation metrics.

1 Introduction

Goal-oriented dialogue systems that can assist humans in various day-to-day activities have widespread applications in several domains such as e-commerce, entertainment, healthcare, and so forth. For example, such systems can help humans in scheduling medical appointments or reserving restaurants, booking tickets. From a modeling perspective, one clear advantage of dealing with domain-specific goal-oriented dialogues is that the vocabulary is typically limited, the utterances largely follow a fixed set of templates, and there is an associated domain knowledge that can be exploited. More specifically, there is some structure associated with the utterances as well as the knowledge base (KB).

More formally, the task here is to generate the next response given (i) the previous utterances in the conversation history, (ii) the current user utterance (known as the query), and (iii) the entities and their relationships in the associated knowledge base. Current state-of-the-art methods (Seo et al., 2017; Eric and Manning, 2017; Madotto et al., 2018) typically use variants of Recurrent Neural Networks (RNNs) (Elman, 1990) to encode the history and current utterance or an external memory network (Sukhbaatar et al., 2015) to encode them along with the entities in the knowledge base. The encodings of the utterances and memory elements are then suitably combined using an attention network and fed to the decoder to generate the response, one word at a time. However, these methods do not exploit the structure in the knowledge base as defined by entity–entity relations and the structure in the utterances as defined by a dependency parse. Such structural information can be exploited to improve the performance of the system, as demonstrated by recent works on syntax-aware neural machine translation (Eriguchi et al., 2016; Bastings et al., 2017; Chen et al., 2017), semantic role labeling (Marcheggiani and Titov, 2017), and document dating (Vashishth et al., 2018), which use Graph Convolutional Networks (GCNs)

Transactions of the Association for Computational Linguistics, vol. 7, pp. 485–500, 2019. https://doi.org/10.1162/tacl_a_00284
Action Editor: Asli Celikyilmaz. Submission batch: 1/2019; Revision batch: 5/2019; Published 9/2019.

(Defferrard et al., 2016; Duvenaud et al., 2015; Kipf and Welling, 2017) to exploit sentence structure.

In this work, we propose to use such graph structures for goal-oriented dialogues. In particular, we compute the dependency parse tree for each utterance in the conversation and use a GCN to capture the interactions between words. This allows us to capture interactions between distant words in the sentence as long as they are connected by a dependency relation. We also use GCNs to encode the entities of the KB where the entities are treated as nodes and their relations as edges of the graph. Once we have a richer structure aware representation for the utterances and the entities, we use a sequential attention mechanism to compute an aggregated context representation from the GCN node vectors of the query, history, and entities. Further, we note that in certain situations, such as when the conversation is in a code-mixed language or a language for which parsers are not available, then it may not be possible to construct a dependency parse for the utterances. To overcome this, we construct a co-occurrence matrix from the entire corpus and use this matrix to impose a graph structure on the utterances. More specifically, we add an edge between two words in a sentence if they co-occur frequently in the corpus. Our experiments suggest that this simple strategy acts as a reasonable substitute for dependency parse trees.

We perform experiments with the modified DSTC2 (Bordes et al., 2017) dataset, which contains goal-oriented conversations for making restaurant reservations. We also use its recently released code-mixed versions (Banerjee et al., 2018), which contain code-mixed conversations in four different languages: Hindi, Bengali, Gujarati, and Tamil. We compare with recent state-of-the-art methods and show that on average, the proposed model gives an improvement of 2.8 BLEU points and 2 ROUGE points. We also perform experiments on two human–human dialogue datasets of different sizes: (i) Cam676 (Wen et al., 2017): a small scale dataset containing 676 dialogues from the restaurant domain; and (ii) MultiWOZ (Budzianowski et al., 2018): a large-scale dataset containing around 10k dialogues and spanning multiple domains for each dialogue. On these two datasets as well, we observe a similar trend, wherein our model outperforms existing methods.

Our contributions can be summarized as follows: (i) We use GCNs to incorporate structural information for encoding query, history, and KB entities in goal-oriented dialogues; (ii) We use a sequential attention mechanism to obtain query aware and history aware context representations; (iii) We leverage co-occurrence frequencies and PPMI (positive-pointwise mutual information) values to construct contextual graphs for code-mixed utterances; and (iv) We show that the proposed model obtains state-of-the-art results on four different datasets spanning five different languages.

2 Related Work

In this section, we review the previous work in goal-oriented dialogue systems and describe the introduction of GCNs in NLP.

Goal-Oriented Dialogue Systems: Initial goal-oriented dialogue systems (Young, 2000; Williams and Young, 2007) were based on dialogue state tracking (Williams et al., 2013; Henderson et al., 2014a,b) and included pipelined modules for natural language understanding, dialogue state tracking, policy management, and natural language generation. Wen et al. (2017) used neural networks for these intermediate modules but still lacked absolute end-to-end trainability. Such pipelined modules were restricted by the fixed slot-structure assumptions on the dialogue state and required per-module based labeling. To mitigate this problem, Bordes et al. (2017) released a version of goal-oriented dialogue dataset that focuses on the development of end-to-end neural models. Such models need to reason over the associated KB triples and generate responses directly from the utterances without any additional annotations. For example, Bordes et al. (2017) proposed a Memory Network (Sukhbaatar et al., 2015) based model to match the response candidates with the multi-hop attention weighted representation of the conversation history and the KB triples in memory. Liu and Perez (2017) further added highway (Srivastava et al., 2015) and residual connections (He et al., 2016) to the memory network in order to regulate the access to the memory blocks. Seo et al. (2017) developed a variant of RNN cell that computes a refined representation of the query over multiple iterations before querying the memory. However, all these approaches retrieve the response from a set of

candidate responses and such a candidate set is not easy to obtain for any new domain of interest. To account for this, Eric and Manning (2017) and Zhao et al. (2017) adapted RNN-based encoder-decoder models to generate appropriate responses instead of retrieving them from a candidate set. Eric et al. (2017) introduced a key-value memory network based generative model that integrates the underlying KB with RNN-based encode-attend-decode models. Madotto et al. (2018) used memory networks on top of the RNN decoder to tightly integrate KB entities with the decoder in order to generate more informative responses. However, as opposed to our work, all these works ignore the underlying structure of the entity–entity graph of the KB and the syntactic structure of the utterances.

GCNs in NLP: Recently, there has been an active interest in enriching existing encode-attend-decode models (Bahdanau et al., 2015) with structural information for various NLP tasks. Such structure is typically obtained from the constituency and/or dependency parse of sentences. The idea is to treat the output of a parser as a graph and use an appropriate network to capture the interactions between the nodes of this graph. For example, Eriguchi et al. (2016) and Chen et al. (2017) showed that incorporating such syntactical structures as Tree-LSTMs in the encoder can improve the performance of neural machine translation. Peng et al. (2017) use Graph-LSTMs to perform cross sentence n-ary relation extraction and show that their formulation is applicable to any graph structure and Tree-LSTMs can be thought of as a special case of it. In parallel, Graph Convolutional Networks (GCNs) (Duvenaud et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2017) and their variants (Li et al., 2016) have emerged as state-of-the-art methods for computing representations of entities in a knowledge graph. They provide a more flexible way of encoding such graph structures by capturing multi-hop relationships between nodes. This has led to their adoption for various NLP tasks such as neural machine translation (Marcheggiani et al., 2018; Bastings et al., 2017), semantic role labeling (Marcheggiani and Titov, 2017), document dating (Vashishth et al., 2018), and question answering (Johnson, 2017; De Cao et al., 2019).

To the best of our knowledge, ours is the first work that uses GCNs to incorporate dependency structural information and the entity–entity graph structure in a single end-to-end neural model for goal-oriented dialogues. This is also the first work that incorporates contextual co-occurrence information for code-mixed utterances, for which no dependency structures are available.

3 Background

In this section, we describe GCNs (Kipf and Welling, 2017) for undirected graphs and then describe their syntactic versions, which work with directed labeled edges of dependency parse trees.

3.1 GCN for Undirected Graphs

Graph convolutional networks operate on a graph structure and compute representations for the nodes of the graph by looking at the neighborhood of the node. We can stack k layers of GCNs to account for neighbors that are k-hops away from the current node. Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of nodes (let $|\mathcal{V}| = n$) and $\mathcal{E}$ is the set of edges. Let $\mathcal{X} \in \mathbb{R}^{n \times m}$ be the input feature matrix with n nodes and each node $\mathbf{x}_u (u \in \mathcal{V})$ is represented by an m-dimensional feature vector. The output of a 1-layer GCN is the hidden representation matrix $\mathcal{H} \in \mathbb{R}^{n \times d}$ where each d-dimensional representation of a node captures the interactions with its 1-hop neighbors. Each row of this matrix can be computed as:

$$\mathbf{h}_v = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (W\mathbf{x}_u + \mathbf{b})\Big), \quad \forall v \in \mathcal{V} \tag{1}$$

Here $W \in \mathbb{R}^{d \times m}$ is the model parameter matrix, $\mathbf{b} \in \mathbb{R}^d$ is the bias vector, and $ReLU$ is the rectified linear unit activation function. $\mathcal{N}(v)$ is the set of neighbors of node v and is assumed to also include the node v so that the previous representation of the node v is also considered while computing its new hidden representation. To capture interactions with nodes that are multiple hops away, multiple layers of GCNs can be stacked together. Specifically, the representation of node v after k^{th} GCN layer can be formulated as:

$$\mathbf{h}_v^{k+1} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (W^k \mathbf{h}_u^k + \mathbf{b}^k)\Big) \tag{2}$$

$\forall v \in \mathcal{V}$. Here $\mathbf{h}_u^k$ is the representation of the u^{th} node in the $(k-1)^{th}$ GCN layer and $\mathbf{h}_u^1 = \mathbf{x}_u$.

3.2 Syntactic GCN

In a directed labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, each edge between nodes u and v is represented by a triple $(u, v, L(u, v))$ where $L(u, v)$ is the associated edge label. Marcheggiani and Titov (2017) modified GCNs to operate over directed labeled graphs, such as the dependency parse tree of a sentence. For such a tree, in order to allow information to flow from head to dependents and vice-versa, they added inverse dependency edges from dependents to heads such as $(v, u, L(u, v)')$ to $\mathcal{E}$ and made the model parameters and biases label specific. In their formulation,

$$\mathbf{h}_v^{k+1} = ReLU\Bigg(\sum_{u \in \mathcal{N}(v)} (W_{L(u,v)}^k \mathbf{h}_u^k + \mathbf{b}_{L(u,v)}^k)\Bigg) \tag{3}$$

$\forall v \in \mathcal{V}$. Notice that unlike equation 2, equation 3 has parameters $W_{L(u,v)}^k$ and $\mathbf{b}_{L(u,v)}^k$ which are label-specific. Suppose there are L different labels, then this formulation will require L weights and biases per GCN layer, resulting in a large number of parameters. To avoid this, the authors use only three sets of weights and biases per GCN layer (as opposed to L) depending on the direction in which the information flows. More specifically, $W_{L(u,v)}^k = W_{dir(u,v)}^k$, where $dir(u, v)$ indicates whether information flows from u to v, v to u or $u = v$. In this work, we also make $\mathbf{b}_{L(u,v)}^k = \mathbf{b}_{dir(u,v)}^k$ instead of having a separate bias per label. The final GCN formulation can thus be described as:

$$\mathbf{h}_v^{k+1} = ReLU\Bigg(\sum_{u \in \mathcal{N}(v)} (W_{dir(u,v)}^k \mathbf{h}_u^k + \mathbf{b}_{dir(u,v)}^k)\Bigg) \tag{4}$$

4 Model

We first formally define the task of end-to-end goal-oriented dialogue generation. Each dialogue of t turns can be viewed as a succession of user utterances (U) and system responses (S) and can be represented as: $(U_1, S_1, U_2, S_2, \ldots, U_t, S_t)$. Along with these utterances, each dialogue is also accompanied by e KB triples that are relevant to that dialogue and can be represented as: $(k_1, k_2, k_3, \ldots, k_e)$. Each triple is of the form: $(entity_1, relation, entity_2)$. These triples can be represented in the form of a graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ where $\mathcal{V}_k$ is the set of all entities and each edge in $\mathcal{E}_k$ is of the form: $(entity_1, entity_2, relation)$, where $relation$ signifies the edge label. At any dialogue turn i, given the (i) dialogue history $H = (U_1, S_1, U_2, \ldots, S_{i-1})$, (ii) the current user utterance as the query $Q = U_i$ and (iii) the associated knowledge graph $\mathcal{G}_k$, the task is to generate the current response S_i that leads to a completion of the goal. As mentioned earlier, we exploit the graph structure in KB and the syntactic structure in the utterances to generate appropriate responses. Toward this end, we propose a model with the following components for encoding these three types of inputs. The code for the model is released publicly.[1]

4.1 Query Encoder

The query $Q = U_i$ is the i^{th} (current) user utterance in the dialogue and contains $|Q|$ tokens. We denote the embedding of the i^{th} token in the query as $\mathbf{q}_i$. We first compute the contextual representations of these tokens by passing them through a bidirectional RNN:

$$\mathbf{b}_t = BiRNN_Q(\mathbf{b}_{t-1}, \mathbf{q}_t) \tag{5}$$

Now, consider the dependency parse tree of the query sentence denoted by $\mathcal{G}_Q = (\mathcal{V}_Q, \mathcal{E}_Q)$. We use a query-specific GCN to operate on $\mathcal{G}_Q$, which takes $\{\mathbf{b}_i\}_{i=1}^{|Q|}$ as the input to the first GCN layer. The node representation in the k^{th} hop of the query specific GCN is computed as:

$$\mathbf{c}_v^{k+1} = ReLU\Bigg(\sum_{u \in \mathcal{N}(v)} (W_{dir(u,v)}^k \mathbf{c}_u^k + \mathbf{g}_{dir(u,v)}^k)\Bigg) \tag{6}$$

$\forall v \in \mathcal{V}_Q$. Here $W_{dir(u,v)}^k, \mathbf{g}_{dir(u,v)}^k$ are edge direction specific query-GCN weights and biases for the k^{th} hop and $\mathbf{c}_u^1 = \mathbf{b}_u$.

4.2 Dialogue History Encoder

The history H of the dialogue contains $|H|$ tokens and we denote the embedding of the i^{th} token in the history by $\mathbf{p}_i$. Once again, we first compute the hidden representations of these tokens using a bidirectional RNN:

$$\mathbf{s}_t = BiRNN_H(\mathbf{s}_{t-1}, \mathbf{p}_t) \tag{7}$$

We now compute a dependency parse tree for each sentence in the history and collectively represent all the trees as a single graph $\mathcal{G}_H =$

[1] https://github.com/sumanbanerjee1/GCN-SeA.

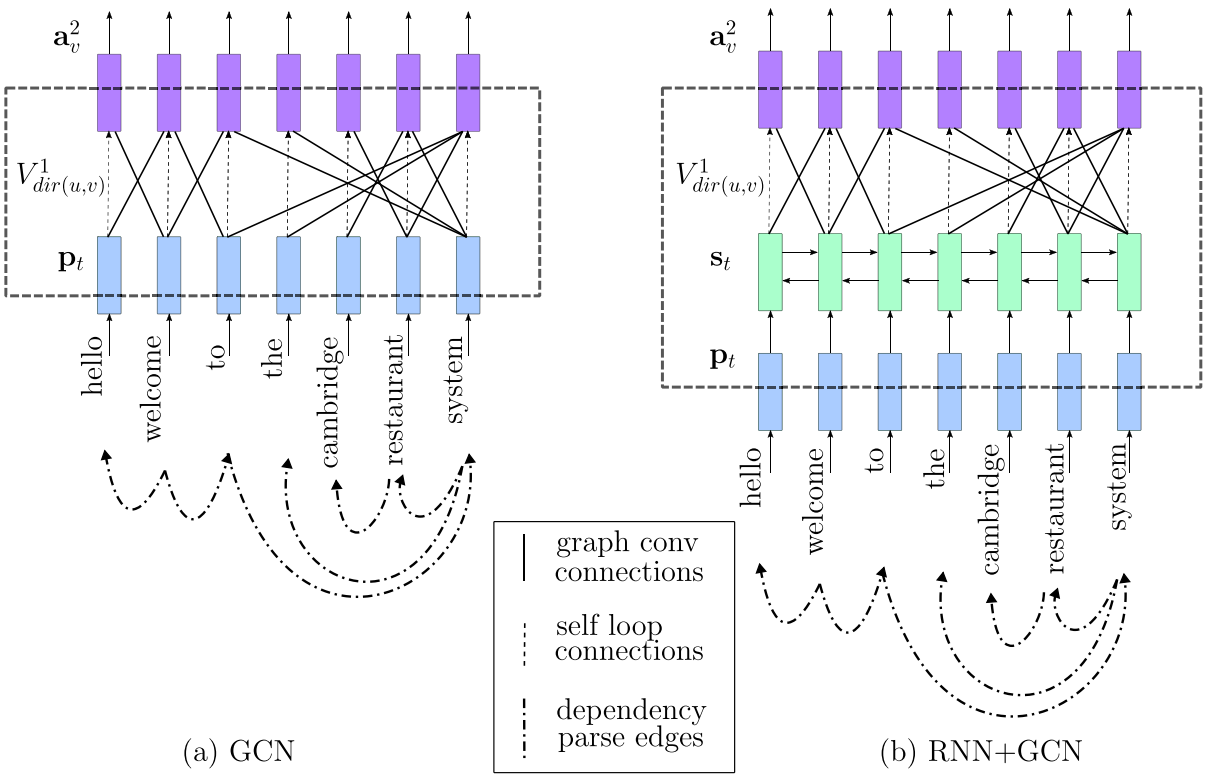


Figure 1: Illustration of the GCN and RNN+GCN modules which are used as encoders in our model. The notations are specific to the dialogue history encoder but both the encoders are similar for the query. We use only the GCN encoder for the KB.

$(\mathcal{V}_H, \mathcal{E}_H)$. Note that this graph will only contain edges between words belonging to the same sentence and there will be no edges between words across sentences. We then use a history-specific GCN to operate on $\mathcal{G}_H$ which takes $\mathbf{s}_t$ as the input to the first layer. The node representation in the k^{th} hop of the history-specific GCN is computed as:

$$\mathbf{a}_v^{k+1} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (V^k_{dir(u,v)}\mathbf{a}_u^k + \mathbf{o}^k_{dir(u,v)})\Big) \tag{8}$$

$\forall v \in \mathcal{V}_H$. Here $V^k_{dir(u,v)}$ and $\mathbf{o}^k_{dir(u,v)}$ are edge direction-specific history-GCN weights and biases in the k^{th} hop and $\mathbf{a}_u^1 = \mathbf{s}_u$. Such an encoder with a single hop of GCN is illustrated in Figure 1(b) and the encoder without the BiRNN is depicted in Figure 1(a).

4.3 KB Encoder

As mentioned earlier, $\mathcal{G}_K = (\mathcal{V}_K, \mathcal{E}_K)$ is the graph capturing the interactions between the entities in the knowledge graph associated with the dialogue. Let there be m such entities and we denote the embedding of the node corresponding to the i^{th} entity as $\mathbf{e}_i$. We then operate a KB-specific GCN on these entity representations to obtain refined representations that capture relations between entities. The node representation in the k^{th} hop of the KB specific GCN is computed as:

$$\mathbf{r}_v^{k+1} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (U^k_{dir(u,v)}\mathbf{r}_u^k + \mathbf{z}^k_{dir(u,v)})\Big) \tag{9}$$

$\forall v \in \mathcal{V}_K$. Here $U^k_{dir(u,v)}$ and $\mathbf{z}^k_{dir(u,v)}$ are edge direction-specific KB-GCN weights and biases in k^{th} hop and $\mathbf{r}_u^1 = \mathbf{e}_u$. We also add inverse edges to $\mathcal{E}_K$ similar to the case of syntactic GCNs in order to allow information flow in both the directions for an entity pair in the knowledge graph.

4.4 Sequential Attention

We use an RNN decoder to generate the tokens of the response and let the hidden states of the decoder be denoted as: $\{\mathbf{d}_i\}_{i=1}^T$, where T is the total number of decoder time steps. In order to obtain a single representation of the node vectors from the final layer ($k = f$) of the query-GCN, we

use an attention mechanism as described below:

$$\mu_{jt} = \mathbf{v}_1^T tanh(W_1 \mathbf{c}_j^f + W_2 \mathbf{d}_{t-1}) \quad (10)$$

$$\boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{\mu}_t) \quad (11)$$

$$\mathbf{h}_t^Q = \sum_{j'=1}^{|Q|} \alpha_{j't} \mathbf{c}_{j'}^f \quad (12)$$

Here $\mathbf{v}_1, W_1$ and W_2 are parameters. Further, at each decoder time step, we obtain a query-aware representation from the final layer of the history-GCN by computing an attention score for each node/token in the history based on the query context vector $\mathbf{h}_t^Q$ as shown below:

$$\nu_{jt} = \mathbf{v}_2^T tanh(W_3 \mathbf{a}_j^f + W_4 \mathbf{d}_{t-1} + W_5 \mathbf{h}_t^Q) \quad (13)$$

$$\boldsymbol{\beta}_t = \text{softmax}(\boldsymbol{\nu}_t) \quad (14)$$

$$\mathbf{h}_t^H = \sum_{j'=1}^{|H|} \beta_{j't} \mathbf{a}_{j'}^f \quad (15)$$

Here $\mathbf{v}_2, W_3, W_4$, and W_5 are parameters. Finally, we obtain a query and history aware representation of the KB by computing an attention score over all the nodes in the final layer of KB-GCN using $\mathbf{h}_t^Q$ and $\mathbf{h}_t^H$ as shown below:

$$\omega_{jt} = \mathbf{v}_3^T tanh(W_6 \mathbf{r}_j^f + W_7 \mathbf{d}_{t-1} + W_8 \mathbf{h}_t^Q + W_9 \mathbf{h}_t^H) \quad (16)$$

$$\boldsymbol{\gamma}_t = \text{softmax}(\boldsymbol{\omega}_t) \quad (17)$$

$$\mathbf{h}_t^K = \sum_{j'=1}^{m} \gamma_{j't} \mathbf{r}_{j'}^f \quad (18)$$

Here $\mathbf{v}_3, W_6, W_7, W_8$ and W_9 are parameters. This sequential attention mechanism is illustrated in Figure 2. For simplicity, we depict the GCN and RNN+GCN encoders as blocks. The internal structure of these blocks are shown in Figure 1.

4.5 Decoder

The decoder is conditioned on two components: (i) the context that contains the history and the KB and (ii) the query that is the last/previous utterance in the dialogue. We use an aggregator that learns the overall attention to be given to the history and KB components. These attention scores: θ_t^H and θ_t^K are dependent on the respective context vectors and the previous decoder state $\mathbf{d}_{t-1}$. The final context vector is obtained as:

$$\mathbf{h}_t^C = \theta_t^H \mathbf{h}_t^H + \theta_t^K \mathbf{h}_t^K \quad (19)$$

$$\mathbf{h}_t^{final} = [\mathbf{h}_t^C; \mathbf{h}_t^Q] \quad (20)$$

where $[;]$ denotes the concatenation operator. At every time step, the decoder then computes a probability distribution over the vocabulary using the following equations:

$$\mathbf{d}_t = RNN(\mathbf{d}_{t-1}, [\mathbf{h}_t^{final}; \mathbf{w}_t]) \quad (21)$$

$$P_{vocab} = \text{softmax}(V' \mathbf{d}_t + \mathbf{b}') \quad (22)$$

where $\mathbf{w}_t$ is the decoder input at time step t, V' and $\mathbf{b}'$ are parameters. P_{vocab} gives us a probability distribution over the entire vocabulary and the loss for time step t is $l_t = -\log P_{vocab}(w_t^*)$, where w_t^* is the t^{th} word in the ground truth response. The total loss is an average of the per-time step losses.

4.6 Contextual Graph Creation

For the dialogue history and query encoder, we used the dependency parse tree for capturing structural information in the encodings. However, if the conversations occur in a language for which no dependency parsers exist, for example: code-mixed languages like Hinglish (Hindi–English) (Banerjee et al., 2018), then we need an alternate way of extracting a graph structure from the utterances. One simple solution that has worked well in practice was to create a word co-occurrence matrix from the entire corpus where the context window is an entire sentence. Once we have such a co-occurrence matrix, for a given sentence we can connect an edge between two words if their co-occurrence frequency is above a threshold value. The co-occurrence matrix can either contain co-occurrence frequency counts or positive-pointwise mutual information (PPMI) values (Church and Hanks, 1990; Dagan et al., 1993; Niwa and Nitta, 1994).

5 Experimental Setup

In this section, we describe the datasets used in our experiments, the various hyperparameters that we considered, and the models that we compared.

5.1 Datasets

The original DSTC2 dataset (Henderson et al., 2014a) was based on the task of restaurant table reservation and contains transcripts of real conversations between humans and bots. The utterances were labeled with the dialogue state annotations like the semantic intent representation, requested slots, and the constraints on the slot values. We report our results on the modified DSTC2 dataset of Bordes et al. (2017), where

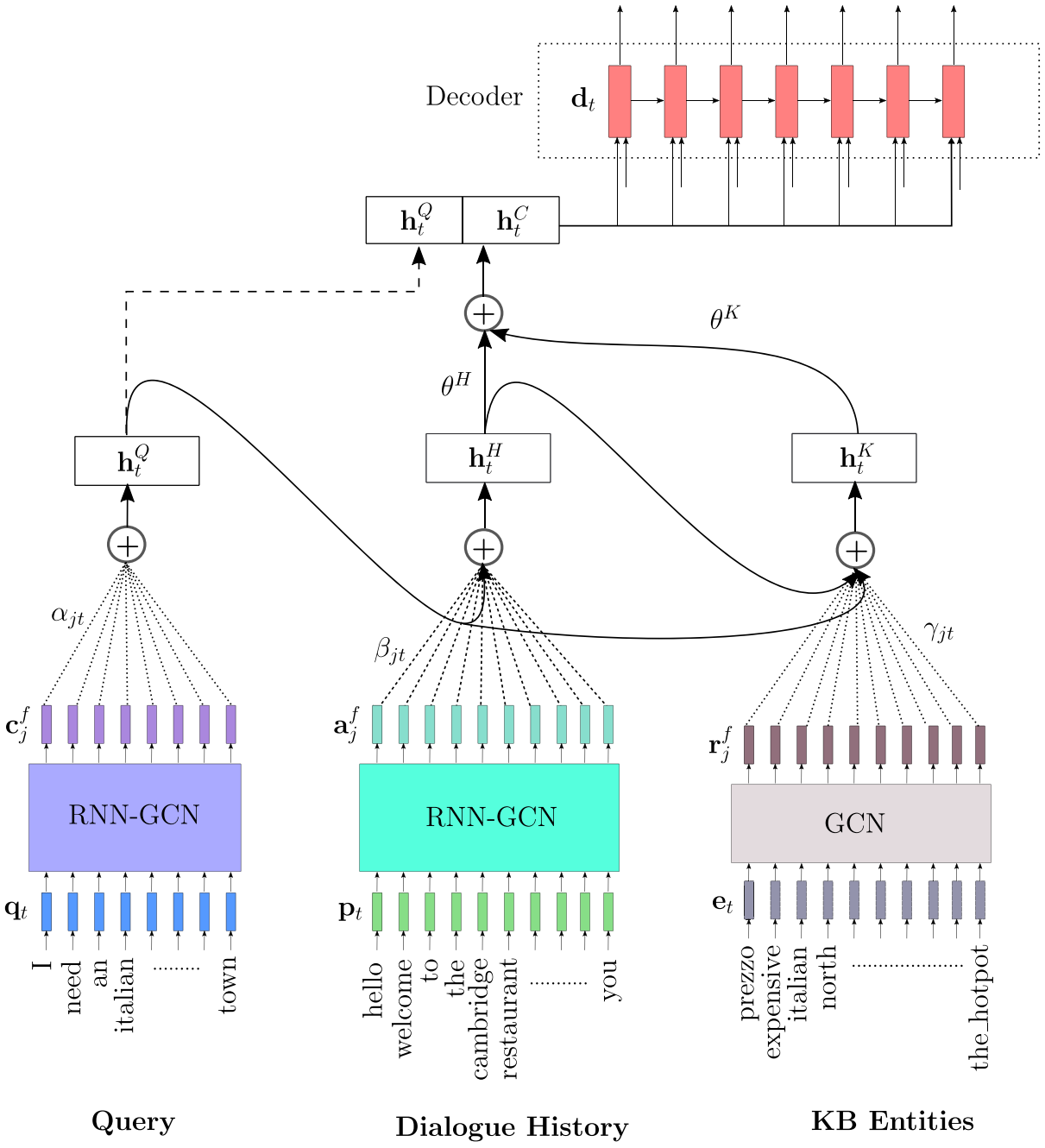


Figure 2: Illustration of sequential attention mechanism in RNN+GCN-SeA.

such annotations are removed and only the raw utterance–response pairs are present with an associated set of KB triples for each dialogue. It contains around 1,618 training dialogues, 500 validation dialogues, and 1,117 test dialogues. For our experiments with contextual graphs we report our results on the code-mixed versions of modified DSTC2, which was recently released by Banerjee et al. (2018). This dataset has been collected by code-mixing the utterances of the English version of modified DSTC2 (En-DSTC2) in four languages: Hindi (Hi-DSTC2), Bengali (Be-DSTC2), Gujarati (Gu-DSTC2), and Tamil (Ta-DSTC2), via crowdsourcing. We also perform experiments on two goal-oriented dialogue datasets that contain conversations between humans wherein the conversations were collected in a Wizard-of-Oz (WOZ) manner. Specifically, we use the Cam676 dataset (Wen et al., 2017), which contains 676 KB-grounded dialogues from the restaurant domain

Model	per-resp. acc	BLEU	ROUGE			Entity F1
			1	2	L	
Rule-Based (Bordes et al., 2017)	33.3	–	–	–	–	–
MEMNN (Bordes et al., 2017)	41.1	–	–	–	–	–
QRN (Seo et al., 2017)	50.7	–	–	–	–	–
GMEMNN (Liu and Perez, 2017)	48.7	–	–	–	–	–
Seq2Seq-Attn (Bahdanau et al., 2015)	46.0	57.3	67.2	56.0	64.9	67.1
Seq2Seq-Attn+Copy (Eric and Manning, 2017)	47.3	55.4	–	–	–	71.6
HRED (Serban et al., 2016)	48.9	58.4	67.9	57.6	65.7	75.6
Mem2Seq (Madotto et al., 2018)	45.0	55.3	–	–	–	75.3
GCN-SeA	47.1	59.0	67.4	57.1	65.0	71.9
RNN+CROSS-GCN-SeA	51.2	60.9	69.4	59.9	67.2	**78.1**
RNN+GCN-SeA	**51.4**	**61.2**	**69.6**	**60.2**	**67.4**	77.9

Table 1: Comparison of RNN+GCN-SeA with other models on the English version of modified DSTC2.

and the MultiWOZ (Budzianowski et al., 2018) dataset, which contains 10,438 dialogues.

5.2 Hyperparameters

We used the same train, test, and validation splits as provided in the original versions of the datasets. We minimized the cross entropy loss using the Adam optimizer (Kingma and Ba, 2015) and tuned the initial learning rates in the range of 0.0006 to 0.001. For regularization we used an L2 penalty of 0.001 in addition to a dropout (Srivastava et al., 2014) of 0.1. We used randomly initialized word embeddings of size 300. The RNN and GCN hidden dimensions were also chosen to be 300. We used GRU (Cho et al., 2014) cells for the RNNs. All parameters were initialized from a truncated normal distribution with a standard deviation of 0.1.

5.3 Models Compared

We compare the performance of the following models.

(i) RNN+GCN-SeA vs GCN-SeA: We use RNN+GCN-SeA to refer to the model described in Section 4. Instead of using the hidden representations obtained from the bidirectional RNNs, we also experiment by providing the token embeddings directly to the GCNs—that is, $\mathbf{c}_u^1 = \mathbf{q}_u$ in equation 6 and $\mathbf{a}_u^1 = \mathbf{p}_u$ in equation 8. We refer to this model as GCN-SeA.

(ii) Cross edges between the GCNs: In addition to the dependency and contextual edges, we add edges between words in the dialogue history/query and KB entities if a history/query word exactly matches the KB entity. Such edges create a single connected graph that is encoded using a single GCN encoder and then separated into different contexts to compute sequential attention. This model is referred to as RNN+CROSS-GCN-SeA.

(iii) GCN-SeA+Random vs GCN-SeA+Structure: We experiment with the model where the graph is constructed by randomly connecting edges between two words in a context. We refer to this model as GCN-SeA+Random. We refer to the model that either uses dependency or contextual graphs instead of random graphs as GCN-SeA+Structure.

6 Results and Discussions

In this section, we discuss the results of our experiments as summarized in Tables 1– 5. We use BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics to evaluate the generation quality of responses. We also report the per-response accuracy, which computes the percentage of responses in which the generated response exactly matches the ground truth response. To evaluate the model's capability of correctly injecting entities in the generated response, we report the entity F1 measure as defined in Eric and Manning (2017).

Results on En-DSTC2: We compare our model with the previous works on the English version of modified DSTC2 in Table 1. For most of the retrieval-based models, the BLEU or ROUGE scores are not available as they select a candidate

Dataset	Model	per-resp. acc	BLEU	ROUGE			Entity F1
				1	2	L	
Hi-DSTC2	Seq2Seq-Bahdanau Attn	48.0	55.1	62.9	52.5	61.0	74.3
	HRED	47.2	55.3	63.4	52.7	61.5	71.3
	Mem2Seq	43.1	50.2	55.5	48.1	54.0	73.8
	GCN-SeA	47.0	56.0	65.0	55.3	63.0	72.4
	RNN+CROSS-GCN-SeA	47.2	56.4	64.7	54.9	62.6	73.5
	RNN+GCN-SeA	**49.2**	**57.1**	**66.4**	**56.8**	**64.4**	**75.9**
Be-DSTC2	Seq2Seq-Bahdanau Attn	**50.4**	55.6	67.4	57.6	65.1	**76.2**
	HRED	47.8	55.6	67.2	57.0	64.9	71.5
	Mem2Seq	41.9	52.1	58.9	50.8	57.0	73.2
	GCN-SeA	47.1	58.4	67.4	57.3	64.9	69.6
	RNN+CROSS-GCN-SeA	**50.4**	59.1	68.3	58.9	65.9	74.9
	RNN+GCN-SeA	50.3	**59.2**	**69.0**	**59.4**	**66.6**	75.1
GU-DSTC2	Seq2Seq-Bahdanau Attn	47.7	54.5	64.8	54.9	62.6	71.3
	HRED	48.0	54.7	65.4	55.2	63.3	71.8
	Mem2Seq	43.1	48.9	55.7	48.6	54.2	**75.5**
	GCN-SeA	48.1	55.7	65.5	56.2	63.5	72.2
	RNN+CROSS-GCN-SeA	**49.4**	**56.9**	**66.4**	**57.2**	**64.3**	73.4
	RNN+GCN-SeA	48.9	56.7	66.1	56.9	64.1	73.0
Ta-DSTC2	Seq2Seq-Bahdanau Attn	49.3	62.9	67.8	56.3	65.6	77.7
	HRED	47.8	61.5	66.9	55.2	64.8	74.4
	Mem2Seq	44.2	58.9	58.6	50.8	57.0	74.9
	GCN-SeA	46.4	62.8	68.5	57.5	66.1	71.9
	RNN+CROSS-GCN-SeA	**50.8**	64.5	69.8	59.6	67.5	**78.8**
	RNN+GCN-SeA	50.7	**64.9**	**70.2**	**59.9**	**67.9**	77.9

Table 2: Comparison of RNN+GCN-SeA with other models on all code-mixed datasets.

Models	Match	Success	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Seq2seq-Attn	85.29	**48.53**	18.81	48.11	24.69	40.41
HRED	83.82	44.12	19.38	48.25	24.09	39.93
GCN-SeA	85.29	21.32	18.48	47.69	25.15	40.29
RNN+GCN-SeA	**94.12**	45.59	**21.62**	**50.49**	**27.69**	**42.35**

Table 3: Comparison of our models with the baselines on the Cam676 dataset.

from a list of candidates as opposed to generating it. Our model outperforms all of the retrieval and generation-based models. We obtain a gain of 0.7 in the per-response accuracy compared with the previous retrieval based state-of-the-art model of Seo et al. (2017), which is a very strong baseline for our generation-based model. We call this a strong baseline because the candidate selection task of this model is easier than the response generation task of our model. We also obtain a gain of 2.8 BLEU points, 2 ROUGE, points and 2.5 entity F1 points compared with current state-of-the-art generation-based models.

Results on code-mixed datasets and effect of using RNNs: The results of our experiments on the code-mixed datasets are reported in Table 2. Our model outperforms the baseline models on all the code-mixed languages. One common observation from the results over all the languages is that RNN+GCN-SeA performs better than GCN-SeA. Similar observations were made by Marcheggiani and Titov (2017) for semantic role labeling.

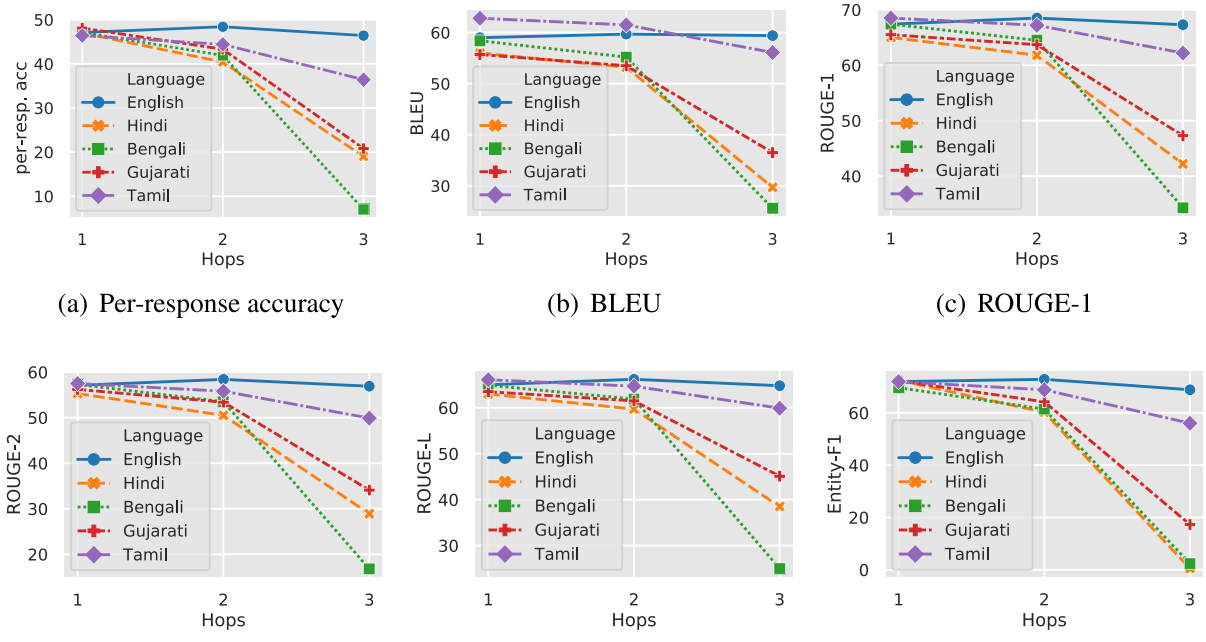
Results on Cam676 dataset: The results of our experiments on the Cam676 dataset are reported in Table 3. In order to evaluate goal-completeness, we use two additional metrics as

Single Domain Dialogues (SNG)						
Models	**Match**	**Success**	**BLEU**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Seq2seq-Attn	68.16	36.77	11.53	35.30	13.44	28.28
HRED	84.30	52.02	10.27	38.30	14.49	30.38
GCN-SeA	63.68	44.84	12.30	39.79	16.11	32.51
RNN+GCN-SeA-400d	**86.10**	**59.19**	11.73	38.76	15.22	30.93
RNN+GCN-SeA-100d	75.78	32.74	**13.13**	**40.76**	**17.67**	**33.59**
Multi-Domain Dialogues (MUL)						
Seq2seq-Attn	44.40	22.10	14.03	38.99	16.39	30.87
HRED	**66.40**	37.70	12.75	40.57	16.83	31.98
GCN-SeA	57.40	37.90	14.16	42.40	19.03	34.25
RNN+GCN-SeA	62.20	**40.30**	**15.85**	**43.40**	**19.63**	**35.15**

Table 4: Comparison of our models with the baselines on the MultiWOZ dataset.

Dataset	Model	per-resp. acc	BLEU	ROUGE 1	ROUGE 2	ROUGE L	Entity F1
En-DSTC2	GCN-SeA+Random	45.9	57.8	67.1	56.5	64.8	**72.2**
	GCN-SeA+Structure	**47.1**	**59.0**	**67.4**	**57.1**	**65.0**	71.9
Hi-DSTC2	GCN-SeA+Random	44.4	54.9	63.1	52.9	60.9	67.2
	GCN-SeA+Structure	**47.0**	**56.0**	**65.0**	**55.3**	**63.0**	**72.4**
Be-DSTC2	GCN-SeA+Random	44.9	56.5	65.4	54.8	62.7	65.6
	GCN-SeA+Structure	**47.1**	**58.4**	**67.4**	**57.3**	**64.9**	**69.6**
Gu-DSTC2	GCN-SeA+Random	45.0	54.0	64.1	54.0	61.9	69.1
	GCN-SeA+Structure	**48.1**	**55.7**	**65.5**	**56.2**	**63.5**	**72.2**
Ta-DSTC2	GCN-SeA+Random	44.8	61.4	66.9	55.6	64.3	70.5
	GCN-SeA+Structure	**46.4**	**62.8**	**68.5**	**57.5**	**66.1**	**71.9**

Table 5: GCN-SeA with random graphs and dependency/contextual graphs on all DSTC2 datasets.



(a) Per-response accuracy (b) BLEU (c) ROUGE-1

(d) ROUGE-2 (e) ROUGE-L (f) Entity F1

Figure 3: GCN-SeA with multiple hops on all DSTC2 datasets.

Dataset	Model	per-resp. acc	BLEU	ROUGE			Entity F1
				1	2	L	
Hi-DSTC2	Seq2seq-Bahdanau Attn	48.0	55.1	62.9	52.5	61.0	74.3
	GCN-Bahdanau Attn	38.5	50.4	58.9	47.7	56.7	59.1
	RNN+GCN-Bahdanau Attn	47.1	56.0	65.1	55.2	62.9	72.2
	RNN-SeA	45.8	55.9	65.1	55.5	63.1	71.8
	RNN+GCN-SeA	**49.2**	**57.1**	**66.4**	**56.8**	**64.4**	**75.9**
Be-DSTC2	Seq2seq-Bahdanau Attn	**50.4**	55.6	67.4	57.6	65.1	**76.2**
	GCN-Bahdanau Attn	42.1	55.1	63.7	52.8	61.1	64.3
	RNN+GCN-Bahdanau Attn	47.0	57.7	67.0	57.4	64.6	70.9
	RNN-SeA	46.8	58.5	67.6	58.1	65.1	71.9
	RNN+GCN-SeA	50.3	**59.2**	**69.0**	**59.4**	**66.6**	75.1
Gu-DSTC2	Seq2seq-Bahdanau Attn	47.7	54.5	64.8	54.9	62.6	71.3
	GCN-Bahdanau Attn	38.8	49.5	59.2	48.3	56.8	58.0
	RNN+GCN-Bahdanau Attn	46.5	55.5	65.6	55.9	63.4	70.6
	RNN-SeA	45.4	56.0	66.0	56.6	63.9	69.8
	RNN+GCN-SeA	**48.9**	**56.7**	**66.1**	**56.9**	**64.1**	**73.0**
Ta-DSTC2	Seq2seq-Bahdanau Attn	49.3	62.9	67.8	56.3	65.6	77.7
	GCN-Bahdanau Attn	42.0	59.3	64.8	52.8	62.1	69.7
	RNN+GCN-Bahdanau Attn	46.3	63.2	68.0	57.2	65.6	72.1
	RNN-SeA	46.8	64.0	69.3	59.0	67.1	74.2
	RNN+GCN-SeA	**50.7**	**64.9**	**70.2**	**59.9**	**67.9**	**77.9**
En-DSTC2	Seq2seq-Bahdanau Attn	46.0	57.3	67.2	56.0	64.9	67.1
	GCN-Bahdanau Attn	45.7	58.1	66.5	55.9	64.1	70.1
	RNN+GCN-Bahdanau Attn	47.4	59.5	67.9	57.7	65.6	72.9
	RNN-SeA	47.0	60.2	68.5	58.9	66.2	72.7
	RNN+GCN-SeA	**51.4**	**61.2**	**69.6**	**60.2**	**67.4**	**77.9**

Table 6: Ablation results of various models on all versions of DSTC2.

Dataset	Model	per resp. acc	BLEU	ROUGE			Entity F1
				1	2	L	
En-DSTC2	Query	22.8	38.1	53.5	37.6	50.6	18.4
	Query + History	47.1	60.6	68.8	59.4	66.6	72.8
	Query + KB	41.4	55.8	63.7	52.4	60.9	63.5
Hi-DSTC2	Query	22.5	37.5	50.9	37.8	48.4	11.1
	Query + History	45.5	55.9	65.3	55.7	63.3	69.8
	Query + KB	40.5	52.6	60.8	49.7	58.5	60.2
Be-DSTC2	Query	22.7	37.9	51.9	38.0	49.0	10.6
	Query + History	45.7	57.4	67.1	57.4	64.6	69.9
	Query + KB	41.2	54.6	63.0	52.1	60.3	60.2
Gu-DSTC2	Query	22.4	36.1	50.7	37.2	48.4	10.9
	Query + History	21.1	36.6	48.6	35.1	46.3	07.2
	Query + KB	40.1	50.6	60.9	50.1	58.7	59.5
Ta-DSTC2	Query	22.8	39.3	53.6	39.0	50.6	18.8
	Query + History	45.8	63.1	68.9	58.4	66.5	72.6
	Query + KB	40.9	59.2	64.2	52.3	61.5	64.2

Table 7: Ablations on different parts of the encoder of RNN+GCN-SeA.

Dataset	Wins %	Losses %	Ties %
En-DSTC2	42.17	22.83	35.00
Cam676	29.00	27.33	43.66

Table 8: Human evaluation results showing wins, losses, and ties % on En-DSTC2 and Cam676.

used in the original paper (Wen et al., 2017) which introduced this dataset, (i) match rate: the number of times the correct entity was suggested by the model, and (ii) success rate: if the correct entity was suggested and the system provided all the requestable slots then the dialogue results in a success. The results suggest that our model's responses are more fluent as indicated by the BLEU and ROUGE scores. It also produces the correct entities according to the dialogue goals but fails to provide enough requestable slots. Note that the model described in the original paper (Wen et al., 2017) is not directly comparable to our work as it uses an explicit belief tracker, which requires extra supervision/annotation about the belief-state. However, for the sake of completeness we would like to mention that their model using this extra supervision achieves a BLEU score of 23.69 and a success rate of 83.82%.

Results on MultiWOZ dataset: The results of our experiments on two versions of the MultiWOZ dataset are reported in Table 4. The first version (SNG) contains around 3K dialogues in which each dialogue involves only a single domain and the second version (MUL) contains all 10k dialogues. The baseline models do not use an oracle belief state as mentioned in Budzianowski et al. (2018) and therefore are comparable to our model. We observed that with a larger GCN hidden dimension (400d in Table 4) our model is able to provide the correct entities and requestable slots in SNG. On the other hand, with a smaller GCN hidden dimension (100d) we are able to generate fluent responses in SNG. On MUL, our model is able to generate fluent responses but struggles in providing the correct entity mainly due to the increased complexity of multiple domains. However, our model still provides a high number of correct requestable slots, as shown by the success rate. This is because multiple domains (*hotel, restaurant, attraction, hospital*) have the same requestable slots (*address, phone, postcode*).

Effect of using hops: As we increased the number of hops of GCNs (Figure 3), we observed a decrease in the performance. One reason for such a drop in performance could be that the average utterance length is very small (7.76 words). Thus, there is not much scope for capturing distant neighborhood information and more hops can add noisy information. The reduction is more prominent in contextual graphs in which multi-hop neighbors can turn out to be dissimilar words in different sentences.

Effect of using random graphs: GCN-SeA+Random and GCN-SeA+Structure take the token embeddings directly instead of passing them though an RNN. This ensures that the difference in performance of the two models are not influenced by the RNN encodings. The results are shown in Table 5 and we observe a drop in performance for GCN-SeA+Random across all the languages. This shows that the dependency and contextual structures play an important role and cannot be replaced by random graphs.

Ablations: We experiment with replacing the sequential attention by the Bahdanau attention (Bahdanau et al., 2015). We also experiment with various combinations of RNNs and GCNs as encoders. The results are shown in Table 6. We observed that GCNs do not outperform RNNs independently. In general, RNN-Bahdanau attention performs better than GCN-Bahdanau attention. The sequential attention mechanism outperforms Bahdanau attention as observed from the following comparisons: (i) GCN-Bahdanau attention vs GCN-SeA, (ii) RNN-Bahdanau attention vs RNN-SeA (in BLEU and ROUGE), and (iii) RNN+GCN-Bahdanau attention vs RNN+GCN-SeA. Overall, the best results are always obtained by our final model, which combines RNN, GCN, and sequential attention. We also performed ablations by removing specific parts of the encoder. Specifically, we experiment with (i) query encoder alone, (ii) query + history encoder, and (iii) query + KB encoder. The results shown in Table 7 suggest that the query and the KB are not enough to generate fluent responses and the previous conversation history is essential.

Human evaluations: In order to evaluate the appropriateness of our model's responses compared to the baselines, we perform a human evaluation of the generated responses using in-house evaluators. We evaluated randomly chosen responses from 200 dialogues of En-DSTC2 and

Context:	**KB Triples**:
Speaker 1: [silence] . **Speaker 2**: hello, welcome to the cambridge restaurant system . you can ask for restaurants by area, price range or food type . how may i help you ? **Speaker 1**: uh cheap restaurant malaysian food . **Speaker 2**: i'm sorry but there is no restaurant serving malaysian food . **Speaker 1**: uh italian food . **Speaker 2**: could you please repeat that ? **Speaker 1**: cheap restaurant uh italian food . **Speaker 2**: there are restaurants serving italian in the cheap price range . what area would you like ? **Speaker 1**: i dont care .	ask, r_cuisine, italian ask, r_location, centre ask, r_phone, ask_phone ask, r_address, ask_address ask, r_price, cheap ask, r_rating, 0 pizza_hut_city_centre, r_cuisine, italian pizza_hut_city_centre, r_location, centre pizza_hut_city_centre, r_phone, pizza_hut_city_centre_phone pizza_hut_city_centre, r_address, pizza_hut_city_centre_address pizza_hut_city_centre, r_price, cheap pizza_hut_city_centre, r_rating, 9
Seq2seq + Attn: ask is a nice place in the west of town serving tasty italian food	
HRED: pizza_hut_city_centre serves italian food in the expensive price range	
GCN-SeA: pizza_hut_city_centre serves italian food in the cheap price range	
RNN+GCN-SeA: pizza_hut_city_centre is a great restaurant serving cheap italian food in the centre of town	
RNN+CROSS-GCN-SeA:pizza_hut_city_centre is a nice place in the centre of town serving tasty italian food	

Table 9: Qualitative comparison of responses between the baselines and different versions of our model

100 dialogues of Cam676 using the method of pairwise comparisons introduced in Serban et al. (2017). We chose the best baseline model for each dataset, namely, HRED for En-DSTC2 and Seq2seq+Attn for Cam676. We show each dialogue context to three different evaluators and ask them to select the most appropriate response in that context. The evaluators were given no information about which model generated which response. They were allowed to choose an option for tie if they were not able to decide whether one model's response was better than the other model. The results reported in Table 8 suggest that our model's responses are favorable in noisy contexts of spontaneous conversations, such as those exhibited in the DSTC2 dataset. However, in a WOZ setting for human–human dialogues, where the conversations are less spontaneous and contexts are properly established, both the models generate appropriate responses.

Qualitative analysis: We show the generated responses of the baselines and different versions of our model in Table 9. We see that Seq2seq+Attn model is not able to suggest a restaurant with a high rating whereas HRED gets the restaurant right but suggests an incorrect price range. However, RNN+GCN-SeA suggests the correct restaurant with the preferred attributes. Although GCN-SeA selects the correct restaurant, it does not provide the location in its response.

7 Conclusion

We showed that structure-aware representations are useful in goal-oriented dialogue and our model outperforms existing methods on four dialogue datasets. We used GCNs to infuse structural information of dependency graphs and contextual graphs to enrich the representations of the dialogue context and KB. We also proposed a sequential attention mechanism for combining the representations of (i) query (current utterance), (ii) conversation history, and (iii) the KB. Finally, we empirically showed that when dependency parsers are not available for certain languages, such as code-mixed languages, then we can use word co-occurrence frequencies and PPMI values to extract a contextual graph and use such a graph with GCNs for improved performance.

Acknowledgments

We would like to thank the anonymous reviewers and the action editor for their insightful comments and suggestions. We would like to thank the Department of Computer Science and Engineering, IIT Madras and Robert Bosch Centre for Data Science and Artificial Intelligence (RBC-DSAI), IIT Madras for providing the necessary resources. We would also like to thank Accenture Technology Labs, India, for supporting our work through their generous academic research grant.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M. Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3766–3780.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967.

Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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, Brussels.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Ido Dagan, Shaul Marcus, and Shaul Markovitch. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 164–171, Columbus, OH.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. 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)*, pages 2306–2317, Minneapolis, MN.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016, Convolutional neural networks on graphs with fast localized spectral filtering, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc.

David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P. Adams. 2015, Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken.

Mihail Eric and Christopher Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th*

Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 468–473.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778, Las Vegas, NV.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 263–272, Philadelphia, PA.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014*, pages 324–329, South Lake Tahoe, NV.

Daniel D. Johnson. 2017. Learning graphical state transitions. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, San Juan.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona.

Fei Liu and Julien Perez. 2017. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017*, pages 1–10, Valencia.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. 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 (Volume 1: Long Papers)*, pages 1468–1478.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515.

Yoshiki Niwa and Yoshihiko Nitta. 1994. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 304–309, Stroudsburg, PA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Query-reduction networks for question answering. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon.

Iulian V. Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, page 1583.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3776–3784, Phoenix, AZ.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015, Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 2440–2448, Montreal.

Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Dating documents using graph convolution networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1615.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia.

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan W. Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 404–413, Metz.

Jason D. Williams and Steve J. Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Steve J. Young. 2000. Probabilistic methods in spoken-dialogue systems. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 358(1769):1389–1402.

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.