GRAPH CONVOLUTIONAL NETWORK WITH SEQUENTIAL ATTENTION FOR GOAL-ORIENTED DIALOGUE SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain specific goal-oriented dialogue systems typically require modeling three types of inputs, *viz.*, (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 and use it to enrich the representations of utterances. We experiment with the modified DSTC2 dataset and its recently released code-mixed versions in four languages and show that our method outperforms existing state-of-the-art methods, using a wide range of evaluation metrics.

1 INTRODUCTION

Goal-oriented dialogue systems which can assist humans in various day-to-day activities have widespread applications in several domains such as e-commerce, entertainment, healthcare, *etc*. For example, such systems can help humans in scheduling medical appointments, reserving restaurants, booking tickets, *etc.*. 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 which can be exploited. More specifically, there is some structure associated with the utterances as well as the knowledge base.

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 relationships in the associated knowledge base. Current state-of-the-art methods (Seo et al., 2017; Eric & Manning, 2017; Madotto et al., 2018) typically use variants of Recurrent Neural Network (Elman, 1990) to encode the history and current utterance and an external memory network to store 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 & Titov, 2017) and document dating

(Vashishth et al., 2018) which use GCNs (Defferrard et al., 2016; Duvenaud et al., 2015; Kipf & 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 the 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 reserving restaurants. We also use its recently released code-mixed versions (Banerjee et al., 2018) which contain code-mixed conversations in four different languages, *viz.*, 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.

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 values to construct contextual graphs for code-mixed utterances and (iv) We show that the proposed model obtains state-of-the-art results on the modified DSTC2 dataset and its recently released code-mixed versions.

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 System : Initial goal-oriented dialogue systems (Young, 2000; Williams & 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 labelling. 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 & 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 which 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 in any new domain of interest. To account for this, Eric & Manning (2017); 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 which 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 to generate more informative responses. However, as opposed to our work, all these works ignore the underlying structure of the entity-relation 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 (NMT). 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 & Welling, 2017) and their variants (Li et al., 2015) 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 & Titov, 2017), document dating (Vashishth et al., 2018) and question answering (Nicola De Cao, 2018).

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 dialogue. 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 Graph Convolutional Networks (GCN) (Kipf & 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 neighbourhood of the node. k layers of GCNs can be stacked to account for neighbours which 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 neighbour. Each row of this matrix can be computed as:

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

Here $W \in \mathbb{R}^{d \times m}$ is the model parameter matrix, $b \in \mathbb{R}^d$ is the bias vector and $ReLU$ is the rectified linear unit activation function. $\mathcal{N}(v)$ is the set of neighbours 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 the new hidden representation. To capture interactions with nodes which 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:

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

where $\mathbf{h}_u^k$ is the representation of the uth 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 & 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,

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

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, $W_{L(u,v)}^k = V_{dir(u,v)}^k$, where $dir(u, v)$ indicates whether information flows from u to v, v to u or $u = v$. In this work, we also make $b_{L(u,v)}^k = b_{dir(u,v)}^k$ instead of having a separate bias per label. The final GCN formulation can thus be described as:

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

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: $(U_1, S_1, U_2, S_2, ..U_t, S_t)$. Along with these utterances, each dialogue is also accompanied by e KB triples which are relevant to that dialogue and can be represented as: $(k_1, k_2, k_3, ..k_e)$. 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 $\mathcal{V}$ is the set of all entities and each edge in $\mathcal{E}$ 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 $H = (U_1, S_1, U_2, ..S_{i-1})$, (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 which 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. Towards this end we propose a model with following components for encoding these three types of inputs.

4.1 QUERY ENCODER

The query $Q = U_i$ is the ith (current) utterance in the dialogue and contains $|Q|$ tokens. We denote the embedding of the ith 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 1st GCN layer. The node representation in the kth hop of the query specific GCN is computed as:

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

where $W_{dir(u,v)}^k, \mathbf{g}_{dir(u,v)}^k$ are edge direction specific query-GCN weights and biases for the kth 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 ith token in the history by $\mathbf{p}_i$ Once again, we first compute the hidden representations of these embeddings 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 $\mathcal{G}_H = (\mathcal{V}_H, \mathcal{E}_H)$. 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 1st layer. The node representation in the kth hop of the history specific GCN is computed as:

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

where $V_{dir(u,v)}^k$ and $\mathbf{o}_{dir(u,v)}^k$ are edge direction specific history-GCN weights and biases in the kth hop and $\mathbf{a}_u^1 = \mathbf{s}_u$.

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 embeddings of the node corresponding to the ith entity as $\mathbf{e}_i$ We then operate a KB specific GCN on these entity representations to obtain refined representations which capture relations between entities. The node representation in the kth hop of the KB specific GCN is computed as:

$$\mathbf{r}_v^{k+1} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (U_{dir(u,v)}^k \mathbf{r}_u^k + \mathbf{z}_{dir(u,v)}^k)\Big) \quad , \quad \forall v \in \mathcal{V}_K \tag{9}$$

where $U_{dir(u,v)}^k$ and $\mathbf{z}_{dir(u,v)}^k$ 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 timesteps. In order to obtain a single representation from the final layer ($k = f$) of the query-GCN node vectors, we use an attention mechanism as described below:

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

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

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

Here $\mathbf{v}_1, W_1, W_2$ are parameters. Further, at each decoder timestep, 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:

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

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

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

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:

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

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

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

Here $\mathbf{v}_3, W_6, W_7, W_8$ and W_9 are parameters.

4.5 DECODER

The decoder takes two inputs, *viz.*, (i) the context which contains the history and the KB and (ii) the query which is the last/previous utterance in the dialogue. We use an aggregator which 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:

$$\mathbf{h}_t^C = \theta_t^H \mathbf{h}_t^H + \theta_t^K \mathbf{h}_t^K \tag{19}$$

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

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

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

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

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 tth 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 which 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 & Hanks, 1990; Dagan et al., 1993; Niwa & Nitta, 1994).

5 EXPERIMENTAL SETUP

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

5.1 DATASETS

The original DSTC2 dataset (Henderson et al., 2014a) was based on the task of restaurant 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 such annotations are removed and only the raw utterance-response pairs are present with an associated set of KB triples for each dialogue. For our experiments with contextual graphs we reported 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 in four languages *viz.* Hindi (Hi-DSTC2), Bengali (Be-DSTC2), Gujarati (Gu-DSTC2) and Tamil (Ta-DSTC2), via crowdsourcing.

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 & Ba, 2015) and tuned the initial learning rates in the range of 0.0006 to 0.001. For regularization we used an L2 norm of

Model	per-resp. acc	BLEU	ROUGE-1	ROUGE-2	ROUGE-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 & Perez, 2017)	48.7	-	-	-	-
Seq2Seq + Attn (Bahdanau et al., 2015)	46.0	57.3	67.2	56.0	64.9
HRED (Serban et al., 2016)	48.9	58.4	67.9	57.6	65.7
Mem2Seq (Madotto et al., 2018)	45.0	55.3	-	-	-
GCN-SeA	47.1	59.0	67.4	57.1	65.0
RNN+GCN-SeA	**51.4**	**61.2**	**69.6**	**60.2**	**67.4**

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

Dataset	Model	per-resp. acc	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Hi-DSTC2	seq2seq + attn	48.0	55.1	62.9	52.5	61.0
	HRED	47.2	55.3	63.4	52.7	61.5
	Mem2Seq	43.1	50.2	55.5	48.1	54.0
	GCN-SeA	47.0	56.0	65.0	55.3	63.0
	RNN+GCN-SeA	**49.2**	**57.1**	**66.4**	**56.8**	**64.4**
Be-DSTC2	seq2seq + attn	**50.4**	55.6	67.4	57.6	65.1
	HRED	47.8	55.6	67.2	57.0	64.9
	Mem2Seq	41.9	52.1	58.9	50.8	57.0
	GCN-SeA	47.1	58.4	67.4	57.3	64.9
	RNN+GCN-SeA	50.3	**59.2**	**69.0**	**59.4**	**66.6**
GU-DSTC2	seq2seq + attn	47.7	54.5	64.8	54.9	62.6
	HRED	48.0	54.7	65.4	55.2	63.3
	Mem2Seq	43.1	48.9	55.7	48.6	54.2
	GCN-SeA	48.1	55.7	65.5	56.2	63.5
	RNN+GCN-SeA	**48.9**	**56.7**	**66.1**	**56.9**	**64.1**
Ta-DSTC2	seq2seq + attn	49.3	62.9	67.8	56.3	65.6
	HRED	47.8	61.5	66.9	55.2	64.8
	Mem2Seq	44.2	58.9	58.6	50.8	57.0
	GCN-SeA	46.4	62.8	68.5	57.5	66.1
	RNN+GCN-SeA	**50.7**	**64.9**	**70.2**	**59.9**	**67.9**

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

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 use 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 i.e. $\mathbf{c}_u^1 = \mathbf{q}_u$ in equation 6 and $\mathbf{a}_u^1 = \mathbf{p}_u$ in equation 8. We refer to this model as GCN-SeA.

(ii) Frequency vs PPMI Contextual Graph : We experiment with the raw frequency co-occurrence graph structure and the PPMI graph structure for the code-mixed datasets, as explained in section 4.6. We refer to these models as GCN-SeA+Freq and GCN-SeA+PPMI. In both these models, the GCN takes inputs from a bidirectional RNN.

(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 which either uses dependency or contextual graph instead of random graphs as GCN-SeA+Structure.

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

Table 3: GCN-SeA with random graphs and frequency co-occurrence graphs on all datasets

6 Results and Discussions

In this section we discuss the results of our experiments as summarized in tables 1,2, and 3. 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.

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 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 to 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 and 2 ROUGE points compared to 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 (including En-DSTC2) is that RNN+GCN-SeA performs better than GCN-SeA. Similar observations were made by Marcheggiani & Titov (2017) for the task of semantic role labeling.

Effect of using Hops: As we increased the number of hops of GCNs, 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 isn't much scope for capturing distant neighbourhood information and more hops can add noisy information. Please refer to Appendix A for detailed results about the effect of varying the number of hops.

Frequency vs PPMI graphs: We observed that PPMI based contextual graphs were slightly better than frequency based contextual graphs (See Appendix B). In particular, when using PPMI as opposed to frequency based contextual graph, we observed a gain of 0.95 in per-response accuracy, 0.45 in BLEU and 0.64 in ROUGE when averaged across all the code-mixed languages.

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 3 and we observe a drop in performance for GCN-Random across all the languages. This shows that any random graph does not contribute to the performance gain of GCN-SeA and the dependency and contextual structures do play an important role.

7 Conclusion

We showed that structure aware representations are useful in goal-oriented dialogue and we obtain state-of-the art performance on the modified DSTC2 dataset and its recently released code-mixed

versions. 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 (ii) 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.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1409.0473.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M. Khapra. A dataset for building code-mixed goal oriented conversation systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3766–3780. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1319.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957–1967. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1209.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *International Conference on Learning Representations*, 2017. URL http://arxiv.org/abs/1605.07683.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 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)*, pp. 1936–1945. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1177. URL http://www.aclweb.org/anthology/P17-1177.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL http://www.aclweb.org/anthology/D14-1179.

Kenneth Ward Church and Patrick Hanks. Word association norms mutual information, and lexicography. *Computational Linguistics*, 16(1), 1990. URL http://www.aclweb.org/anthology/J90-1003.

Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. In *31st Annual Meeting of the Association for Computational Linguistics*, 1993. URL http://www.aclweb.org/anthology/P93-1022.

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

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

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

Mihail Eric and Christopher Manning. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 468–473. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/E17-2075.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pp. 37–49, 2017. URL https://aclanthology.info/papers/W17-5506/w17-5506.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 823–833. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1078. URL http://www.aclweb.org/anthology/P16-1078.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 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, 18-20 June 2014, Philadelphia, PA, USA*, pp. 263–272, 2014a. URL http://aclweb.org/anthology/W/W14/W14-4337.pdf.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, pp. 324–329, 2014b. doi: 10.1109/SLT.2014.7078595. URL https://doi.org/10.1109/SLT.2014.7078595.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

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

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. *CoRR*, abs/1511.05493, 2015. URL http://arxiv.org/abs/1511.05493.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. URL http://www.aclweb.org/anthology/W04-1013.

Fei Liu and Julien Perez. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pp. 1–10, 2017. URL https://aclanthology.info/papers/E17-1001/e17-1001.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1468–1478. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1136.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506–1515. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1159.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. 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), pp. 486–492. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-2078.

Ivan Titov Nicola De Cao, Wilker Aziz. Question answering by reasoning across documents with graph convolutional networks. *arXiv preprint arXiv:1808.09920*, 2018.

Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pp. 304–309, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/991886.991938. URL https://doi.org/10.3115/991886.991938.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pp. 311–318, 2002. URL http://www.aclweb.org/anthology/P02-1040.pdf.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017. ISSN 2307-387X. URL https://www.transacl.org/ojs/index.php/tacl/article/view/1028.

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

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pp. 3776–3784, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. URL http://dl.acm.org/citation.cfm?id=2670313.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2440–2448, 2015. URL http://papers.nips.cc/paper/5846-end-to-end-memory-networks.

Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Dating documents using graph convolution networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1605–1615. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1149.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 438–449. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/E17-1042.

Jason D. Williams and Steve J. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007. doi: 10.1016/j.csl.2006.06.008. URL https://doi.org/10.1016/j.csl.2006.06.008.

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan W. Black. 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, 22-24 August 2013, SUPELEC, Metz, France, pp. 404–413, 2013. URL http://aclweb.org/anthology/W/W13/W13-4065.pdf.

Steve J. Young. Probabilistic methods in spoken-dialogue systems. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 358(1769):1389–1402, 2000. ISSN 1364503X. URL http://www.jstor.org/stable/2666825.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 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*, pp. 27–36. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-5505.

A. EFFECT OF USING HOPS:

Dataset	Model	per-resp. acc	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
En-DSTC2	GCN-SeA+K=1	47.1	59.0	67.4	57.1	65.0
	GCN-SeA+K=2	**48.4**	**59.7**	**68.5**	**58.4**	**66.2**
	GCN-SeA+K=3	46.4	59.4	67.3	56.9	64.8
Hi-DSTC2	GCN-SeA+K=1	**47.0**	**56.0**	**65.0**	**55.3**	**63.0**
	GCN-SeA+K=2	40.4	53.2	61.8	50.5	59.7
	GCN-SeA+K=3	19.0	29.7	42.2	28.9	38.5
Be-DSTC2	GCN-SeA+K=1	**47.1**	**58.4**	**67.4**	**57.3**	**64.9**
	GCN-SeA+K=2	41.9	55.2	64.5	53.5	61.9
	GCN-SeA+K=3	7.0	25.6	34.3	16.8	25.0
GU-DSTC2	GCN-SeA+K=1	**48.1**	**55.7**	**65.5**	**56.2**	**63.5**
	GCN-SeA+K=2	43.3	53.5	63.7	53.4	61.5
	GCN-SeA+K=3	20.8	36.5	47.3	34.1	45.1
Ta-DSTC2	GCN-SeA+K=1	**46.4**	**62.8**	**68.5**	**57.5**	**66.1**
	GCN-SeA+K=2	44.4	61.5	67.2	55.8	64.7
	GCN-SeA+K=3	36.4	56.1	62.2	49.9	59.9

Table 4: GCN-SeA with multiple hops on all datasets

B. FREQUENCY VS PPMI CO-OCCURRENCE

Dataset	Model	per-resp. acc	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
En-DSTC2	GCN-SeA+Freq	50.4	**61.1**	69.3	59.6	67.0
	GCN-SeA+PPMI	**50.5**	60.7	69.3	**59.7**	67.0
Hi-DSTC2	GCN-SeA+Freq	48.7	56.9	65.5	56.1	63.5
	GCN-SeA+PPMI	**49.2**	**57.1**	**66.4**	**56.8**	**64.4**
Be-DSTC2	GCN-SeA+Freq	49.0	59.0	68.2	58.5	65.7
	GCN-SeA+PPMI	**50.3**	**59.2**	**69.0**	**59.4**	**66.6**
Gu-DSTC2	GCN-SeA+Freq	48.4	56.1	**66.2**	56.7	64.0
	GCN-SeA+PPMI	**48.9**	**56.7**	66.1	**56.9**	**64.1**
Ta-DSTC2	GCN-SeA+Freq	49.2	64.1	69.5	59.0	67.1
	GCN-SeA+PPMI	**50.7**	**64.9**	**70.2**	**59.9**	**67.9**

Table 5: RNN+GCN-SeA with different contextual graphs on all datasets