

# RDF-to-Text Generation with Graph-augmented Structural Neural Encoders

Hanning Gao<sup>1 \*</sup>, Lingfei Wu<sup>2 \*</sup>, Po Hu<sup>1,3 †</sup> and Fangli Xu<sup>4</sup>

<sup>1</sup>School of Computer Science, Central China Normal University, Wuhan, China

<sup>2</sup>IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA

<sup>3</sup>Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Wuhan, China

<sup>4</sup>Squirrel AI Learning

gaohn@mail.ccnu.edu.cn, lwu@email.wm.edu, phu@mail.ccnu.edu.cn, lili@yixue.us

## Abstract

The task of RDF-to-text generation is to generate a corresponding descriptive text given a set of RDF triples. Most of the previous approaches either cast this task as a sequence-to-sequence problem or employ graph-based encoder for modeling RDF triples and decode a text sequence. However, none of these methods can explicitly model both local and global structure information between and within the triples. To address these issues, we propose to jointly learn local and global structure information via combining two new graph-augmented structural neural encoders (i.e., a bidirectional graph encoder and a bidirectional graph-based meta-paths encoder) for the input triples. Experimental results on two different WebNLG datasets show that our proposed model outperforms the state-of-the-art baselines. Furthermore, we perform a human evaluation that demonstrates the effectiveness of the proposed method by evaluating generated text quality using various subjective metrics.

## 1 Introduction

RDF-to-text generation is to transform a set of Resource Description Framework (RDF) triples into informative and faithful text. The task is challenging since RDF sub-graph structure needs to be well modeled in addition to capturing semantic information in RDF triples. Figure 1 illustrates an RDF triples graph with corresponding descriptive text, in which the nodes (such as "FOOD-1" and "INGREDIENT") represent the entities and edges (such as "dishVariation" and "ingredient") represent the relations between the connected entities. It has many fact-aware applications such as knowledge-based question answering [Hao *et al.*, 2017], entity summarization [Pouriyeh *et al.*, 2017], and data-driven text generation [Liu *et al.*, 2018].

Traditionally, an RDF-to-text generation system mainly focuses on the pipeline process of content selection and surface realization with human-crafted features. However, a

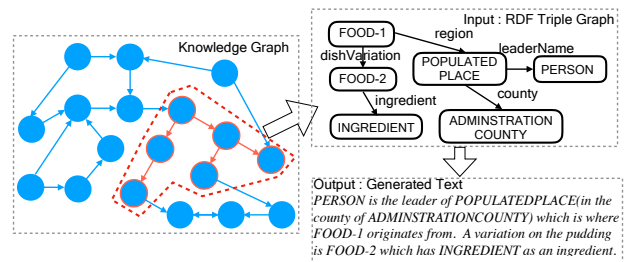


Figure 1: A knowledge graph formulated by a set of RDF triples with generated text descriptions.

critical issue for error propagation has been largely overlooked, which does harm to the quality of generated text. Recently, as the end-to-end deep learning has made great progress in natural language processing, RDF-to-text generation has achieved promising performance by using various sequence-to-sequence (Seq2Seq) models [Gardent *et al.*, 2017b; Jagfeld *et al.*, 2018]. Conceptually, RDF triple elements need to be processed and concatenated into a sequence in order to feed into Seq2Seq models.

However, simply transforming the RDF triples into a sequence may lose important higher-order information. Since RDF triples can be represented as a knowledge graph, two graph-based approaches have been proposed recently for RDF-to-text generation. Trisedya *et al.* presented a graph-based triple encoder GTR-LSTM that captures both intra-triple and inter-triple entity relationships by sampling different meta-paths to preserve graph structure in the encoder. However, since the encoder component is still based on recurrent neural networks, it often fails to capture rich local complex structure information between entities and relationships.

On the other hand, Marcheggiani and Perez-Beltrachini proposed a graph-to-sequence model (Graph2Seq) based on a modified Graph Convolutional Networks (GCN) [Kipf and Welling, 2016], which directly encodes graph-structured RDF triples and decodes a text sequence. However, it is well known that GCN often gets overfitting quickly when using multiple layers ( $\geq 3$ ), which weakens its capability of learning longer range dependence. Therefore, this model usually performs better on capturing local structure information of the graph than capturing global information between the

\*Both authors contributed equally to this research.

†Corresponding author.

RDF triples.

To address the aforementioned issues, we propose a novel neural network architecture by exploiting graph-augmented structural neural encoders for RDF-to-text generation. To this end, we first present to exploit both power of graph encoder and graph-based meta-paths encoder to jointly learn structure information locally and globally. We then present separated attentions on two different inputs i.e., graphs and meta-paths to jointly learn the final hidden representation of RDF triples in order to better decode the text sequence.

The combined encoders can focus on multiple perspectives of the input RDF graph. A novel bidirectional-GCN (bi-GCN) encoder is used to explicitly model the local structure information between the intra-triple relationships, while a new bidirectional Graph-based Meta-Paths (bi-GMP) encoder mainly focuses on modeling global long-range dependency between the inter-triple relationships.

We highlight our main contributions as follows:

- We propose a novel graph-augmented structural neural encoders model by combining a new bi-GCN encoder and a new bi-GMP encoder for explicitly modeling the global and local structure information of the input RDF triples.
- We further present separated attentions on each of graph encoders and fuse their corresponding context vectors to better decode the descriptive text.
- The experimental results on two WebNLG datasets (a challenge dataset and a supplementary dataset) corroborate the advantages of our model over state-of-the-art models in BLEU, METEOR and TER metrics.

## 2 Related Work

Our approach is highly related with existing works in structured data to text generation and graph neural networks.

### 2.1 RDF-to-Text Generation

RDF-to-text generation aims to generate a grammatically correct, fluent, informative and faithful description for graph structured input data. Early attempts for Web Ontology Language to text generation were done by [Bontcheva and Wilks, 2004] and verbalizing a knowledge base [Banik *et al.*, 2012]. Pipeline approach is generally used to solve these data-to-text generation tasks, which includes two main steps: (1) *content selection* [Barzilay and Lapata, 2005] defines what contents should be described in the generated text; (2) *surface realization* [Deemter *et al.*, 2005] implements the generation process word by word. In the past few years, diverse NLG tasks have achieved promising performance by using Seq2Seq model with attention mechanism [Bahdanau *et al.*, 2014] and copy mechanism [Gu *et al.*, 2016; See *et al.*, 2017]. As shown in [Gardent *et al.*, 2017b; Jagfeld *et al.*, 2018], Seq2Seq model and its variants perform promisingly on RDF-to-text generation by concatenating RDF triple elements into a sequence. To reduce the probability of generating low-quality sentences, Zhu *et al.* proposed a framework based on Seq2Seq, which optimizes the inverse Kullback-Leibler (KL) divergence between the distributions of the real and generated sentences. By combining

the pipeline system and neural networks, Moryossef *et al.* presented a method for matching reference texts to their corresponding text plans to train a plan-to-text generator.

### 2.2 Graph Neural Networks for Text Generation

Recently, there has been a surge of interests to exploit graph-based neural networks for graph-structured data to text generation. Researchers employed a bidirectional graph encoder to embed an input graph to a sequence of node embeddings and then used an attention-based LSTM method to decode the target sequence from these vectors [Xu *et al.*, 2018; Chen *et al.*, 2020; Gao *et al.*, 2019]. Song *et al.* applied an LSTM over the state transitions of the graph-based encoder outputs to capture longer-range dependencies. Marcheggiani and Perez-Beltrachini exploited relational graph convolutional networks, introduced and extended in [Bruna *et al.*, 2013; Defferrard *et al.*, 2016; Kipf and Welling, 2016], to encode a node in a relational graph with its neighbor nodes, edge labels and edge directions at the same time. Although these graph neural networks consider different perspectives of graph structure information, none of graph-based encoders could fully capture both global and local dependencies in a knowledge graph.

## 3 Problem Definition

Formally, the RDF-to-text generation task is defined as follows. The input contains a set of RDF triples, denoted as  $S = \{t_1, t_2, \dots, t_n\}$  where  $t_i$  is a triple consisting of subject, relationship and object. All the input triples are represented as directed graphs. The aim is to generate a natural language descriptive text  $Y = \langle w_1, w_2, \dots, w_T \rangle$  that represents the correct and concise semantics of entities and their relationships in the given RDF triple inputs. In this study, the bi-GMP encoder consumes input graph  $G_1 = (V_1, E_1)$ , where  $V_1$  represents all of the entity nodes and  $E_1$  denotes original relationships between these entity nodes. Differently, the bi-GCN encoder consumes input graph  $G_2 = (V_2, E_2)$ , where  $V_2$  contains entity nodes and relationship nodes since relationships are regarded as additional nodes instead of edges.  $E_2$  refers to a newly defined set of edges that describe the relationship between entity nodes and relationship nodes, or between multiple words of a node.

## 4 Our Proposed Model

In this section, we present our graph-augmented structural neural encoders including two graph-based encoders, and describe the approach to combine them for better capturing global and local relationships between and within the triples.

### 4.1 Bidirectional Graph-based Meta-Paths Encoder

In order to encode information according to different meta-paths and capture long-range dependency in graph  $G_1$ , we present a new bidirectional Graph-based Meta-Paths encoder (bi-GMP), which is an improved variant of GTR-LSTM triple encoder [Trisedya *et al.*, 2018]. Compared to GTR-LSTM, the bi-GMP model applies hidden state masking between different meta-paths to keep the paths encoding non-interfering.

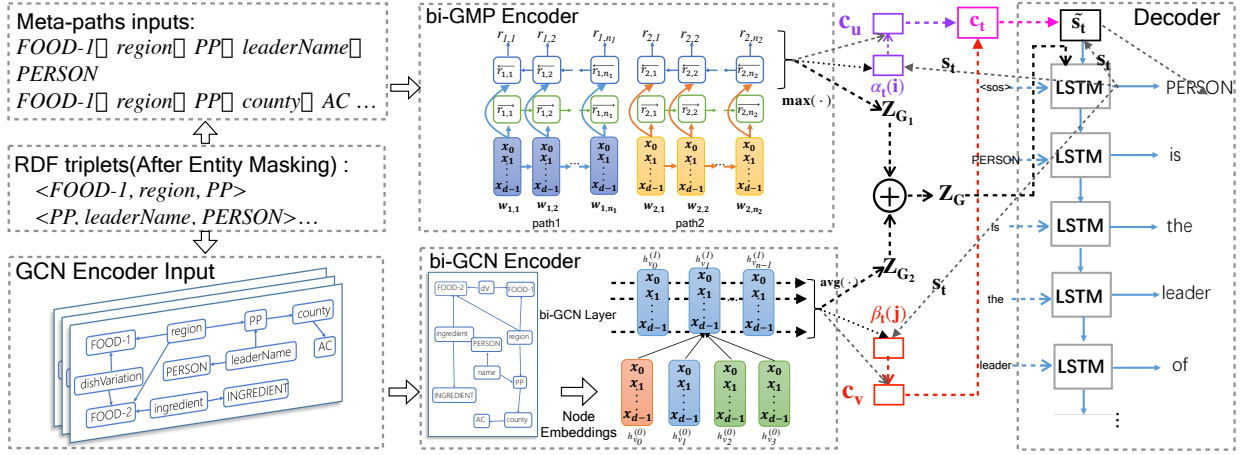


Figure 2: The framework of the combined Graph-augmented Structural Neural Encoders Model.

Moreover, copy mechanism [See *et al.*, 2017] is introduced into our meta-paths encoder to improve the performance.

The meta-paths of graph  $G_1$  are calculated by a combination algorithm of topological sort and single-source shortest path. We first calculate the nodes with zero in-degree  $V_{IN}$  and nodes with zero out-degree  $V_{OUT}$ . Then, we calculate a single-source shortest path using a node in  $V_{IN}$  as the source node and a node in  $V_{OUT}$  as the destination node with both the entity nodes and the relationships preserved in the path. The graph  $G_1$  is transformed to a sequence composed of a set of meta-paths  $S_p = \{p_1, p_2, \dots\}$ , in which  $p_k = \langle w_{k,1}, w_{k,2}, \dots, w_{k,n_k} \rangle$ .

Inspired by the advantages of BiLSTM over LSTM, the proposed meta-paths encoder calculates the representation of each token in each meta-path from two directions and fuse them together in the final step. The hidden state  $r_i$  is computed as:

$$\vec{r}_i = f(\overleftarrow{r}_{i-1}, w_i) \text{ and } \overleftarrow{r}_i = g(\overleftarrow{r}_{i+1}, w_i) \quad (1)$$

$$r_i = \text{CONCAT}(\vec{r}_i, \overleftarrow{r}_i) \quad (2)$$

where  $w_i$  represents a token of entity node or relationship.  $f(\cdot)$  and  $g(\cdot)$  are single LSTM units. Finally, the bi-GMP encoder generates a set of entity node representations  $R_1$ . We then use max pooling on  $R_1$  to compute the graph embedding of graph  $G_1$ :  $Z_{G_1} = W_r \text{maxpool}(R_1)$ , where  $W_r$  denotes weight matrices.

## 4.2 Bidirectional Graph Convolutional Networks Encoder

For the graph  $G_2 = (V_2, E_2)$ , relationships are regarded as additional graph nodes, so  $G_2$  contains entity nodes and relationship nodes. Then, for the similar reason above, we present the bi-GCN encoder calculating the vector representations  $H^{(l)} = \{h_{v_0}^{(l)}, h_{v_1}^{(l)}, \dots\} \in \mathbb{R}^{D \times V}$  at layer  $l$ :

$$H_+^{(l)} = \hat{D}_+^{-1/2} \hat{A}_+ \hat{D}_+^{-1/2} H^{(l-1)} W_{1+}^{(l-1)} \quad (3)$$

$$H_-^{(l)} = \hat{D}_-^{-1/2} \hat{A}_- \hat{D}_-^{-1/2} H^{(l-1)} W_{1-}^{(l-1)} \quad (4)$$

$$H^{(l)} = \sigma(\text{CONCAT}(H_+^{(l)}, H_-^{(l)}) W_f) \quad (5)$$

where  $\hat{A}_+ = A_+ + I$  and  $\hat{A}_- = A_- + I$  denote the source-to-target and target-to-source adjacency matrices of the directed graph  $G_2$  with inserted self-loops.  $I$  is an identity matrix and  $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$  is a diagonal degree matrix.  $W_{1+}^{(l-1)}$  and  $W_{1-}^{(l-1)}$  are layer-specific trainable weight matrices.  $W_f$  denotes trainable weight matrix, and  $\sigma$  is a non-linearity function.  $H^{(0)}$  is initialized with word embeddings. The bidirectional node embeddings at layer  $l$  are concatenated and input to a single-layer perceptron before input to the next bi-GCN layer. Bi-GCN encoder is stacked to  $L$  layer and  $R_2 = H^{(L)}$  is a set of entity and relationship node representations.

We then use average pooling on  $R_2$  to compute the graph embedding of graph  $G_2$ :  $Z_{G_2} = \varphi(\text{avgpool}(R_2))$ , where  $\varphi(\cdot)$  is a single-layer perceptron.

## 4.3 Combining bi-GMP and bi-GCN Encoders

We propose a combination strategy to integrate the above bi-GMP encoder and the bi-GCN encoder for the input RDF triples, which aims to jointly learn the local and global structure information of the RDF triples input. The overall architecture of the combined encoders is depicted in Fig. 2.

Both encoders generate a set of node representations. In general,  $R_2 = \{h_0, h_1, h_2, \dots\}$  captures the local structure information within the RDF triples better, since each node representation is directly modeled by all its one-hop neighbors at one layer. To avoid overfitting, bi-GCN encoder is stacked to 2 layers in this study. In this way, the output of bi-GCN encoder  $h_i$  preserves at most two-hop information of the graph, which is limited to one or two triples. Whilst,  $R_1 = \{r_0, r_1, r_2, \dots\}$  mainly focuses on the global structure information between the RDF triples, since the bi-GMP encoder computes hidden states following a traversal order calculated by a combination of topological sort and single-source shortest path algorithm over the whole graph. The output of the bi-GMP encoder  $r_i$  can preserve information from one triple to seven triples (an input contains at most seven triples in the WebNLG datasets) if there exists a meta-path.

Then, the combined graph embedding  $Z_G$  is computed as

$Z_G = Z_{G_1} \oplus Z_{G_2}$ , where  $\oplus$  is the component-wise addition.

#### 4.4 Decoder

An attention-based LSTM decoder [Luong *et al.*, 2015] is used for text generation. However, the plain top-down attention is not appropriate since we have two input graphs and it can not fully exploit different semantic token information from two quite different encoders. Therefore, we present separated attention mechanism on each of the graph encoders and fuse their corresponding context vectors to better decode the descriptive text.

$Z_G$  is fed into the decoder as initial hidden state. For each time step  $t$ , the decoder feeds the concatenation of the embedding of the current input (previously generated word)  $e_t$  and previous step context vector  $c_{t-1}$  as new input and previous hidden state  $s_{t-1}$  to update its hidden state  $s_t$ . Then, we further apply the separate attentions by computing the align weights vectors at decoding time step  $t$  as follows:

$$\alpha_{t(i)} = \frac{\exp(\text{score}(r_i, s_t))}{\sum_{k=1}^M \exp(\text{score}(r_k, s_t))} \quad (6)$$

$$\beta_{t(j)} = \frac{\exp(\text{score}(h_j, s_t))}{\sum_{k=1}^V \exp(\text{score}(h_k, s_t))} \quad (7)$$

where  $r_i \in R_1$  and  $h_j \in R_2$ .  $M = |R_1|$  and  $V = |R_2|$  are the lengths of representations sequence.  $s_t$  is the hidden state of decoder. The  $\text{score}(\cdot)$  function estimates the similarity of  $r_i, h_j$  and  $s_t$ .

Then, we compute bi-GMP level context vector  $c_u$  and bi-GCN level context vector  $c_v$  respectively:

$$c_u = \sum_{i=1}^M \alpha_{t(i)} r_i \quad \text{and} \quad c_v = \sum_{j=1}^V \beta_{t(j)} h_j \quad (8)$$

Next, we concatenate  $c_u, c_v$  and decoder hidden state  $s_t$  to compute final attentional hidden state at this time step as:

$$c_t = \text{CONCAT}(c_u, c_v) \quad (9)$$

$$\tilde{s}_t = \tanh(W_c \cdot [c_t; s_t] + b_c) \quad (10)$$

where  $c_t$  is the concatenated context vector at time step  $t$  and  $W_c$  and  $b_c$  are learnable parameters. After calculating a sequence of attentional hidden states  $\langle \tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_T \rangle$  recurrently, the decoder generates an output sequence  $\langle y_1, y_2, \dots, y_T \rangle$ . The output probability distribution over a vocabulary at the current time step is calculated by:

$$P(y_t | y_{1:t-1}) = \text{softmax}(W_v \tilde{s}_t + b_v) \quad (11)$$

where  $W_v$  and  $b_v$  are learnable parameters. Finally, we employ the loss function with a negative log likelihood:

$$L_G = \frac{1}{T} \sum_{t=1}^T -\log P(y_t | y_{1:t-1}) \quad (12)$$

## 5 Entity Masking and Graph Constructions

In this section, we first discuss the importance of entity masking and then describe how to transform RDF triples set to input graphs for two encoders.

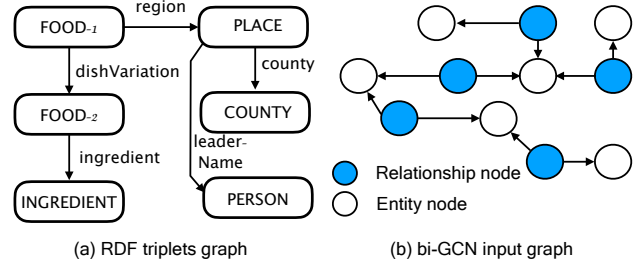


Figure 3: Examples of two graph inputs.

### 5.1 Entity Masking

Entity masking can improve the generalization ability of our model. We use the officially provided dictionary and DBpedia lookup API to map the subjects and objects in the triples set to corresponding entity types. Considering that multiple entities in a set of triples may belong to the same type, we assign an entity-id (eid) to each entity in the set. Therefore, each entity is replaced with its eid and type. For example, the entity ‘‘Bakewell pudding’’, namely FOOD-1 in Fig. 3, is replaced by ENTITY-1 FOOD while the entity ‘‘Bakewell tart’’ (FOOD-2) is replaced by ENTITY-2 FOOD.

### 5.2 RDF Triples to Meta-paths

In Fig. 3(a), we first choose a source node with zero in-degree FOOD-1 and a destination node with zero out-degree PERSON. The shortest path from FOOD-1 to PERSON is FOOD-1  $\rightarrow$  region  $\rightarrow$  PLACE  $\rightarrow$  leaderName  $\rightarrow$  PERSON. Similarly, the other two meta-paths are:

- FOOD-1  $\rightarrow$  region  $\rightarrow$  PLACE  $\rightarrow$  county  $\rightarrow$  COUNTY
- FOOD-1  $\rightarrow$  dishVariation  $\rightarrow$  FOOD-2  $\rightarrow$  ingredient  $\rightarrow$  INGREDIENT

The meta-paths are concatenated to a sequence and then fed to the bi-GMP encoder. The hidden state of the last token in path  $p_t$  is not forwarded to the first token in path  $p_{t+1}$ , which is depicted in Fig. 2.

### 5.3 RDF Triples to bi-GCN Graph

For bi-GCN encoder, we treat relationships as additional nodes similar to [Marcheggiani and Titov, 2017] and the new relationship node is connected to the subject and object by two new directed edges, respectively. This graph construction method makes the new bi-GCN graph (Fig. 3(b)) two times larger than the original RDF triples graph (Fig. 3(a)) in hops, making it more difficult to capture long-range dependency. As shown in Fig. 3(b), the nodes in the graph are divided into entity nodes and relationship nodes. The original triple (FOOD-1, region, PLACE) is separated into two new triples region  $\rightarrow$  FOOD-1 and region  $\rightarrow$  PLACE with ‘‘region’’ being a node in the input graph. For nodes having more than one word, each word is separated into an independent node that is connected to the core node (usually the ‘‘ENTITY-id’’ node) with a new edge. In this way, the original entity ‘‘ENTITY-1 FOOD INGREDIENTS’’ is separated into two new triples ENTITY-1  $\rightarrow$  FOOD and ENTITY-1  $\rightarrow$  INGREDIENTS.

Model	multi-BLEU $\uparrow$	METEOR $\uparrow$	TER $\downarrow$
BiLSTM	53.69	40.5	43.7
MELBOURNE	54.52	41.0	<b>40.0</b>
PKUWRITER	51.23	37.0	45.0
DGCN	55.90	39.0	41.0
bi-GMP(copy)	55.32	41.1	41.7
bi-GCN(2L)	55.85	42.2	41.1
bi-GCN(3L)	55.32	41.8	41.0
bi-GCN(2L)+copy	55.71	40.9	40.9
biGCN(2L) + bi-GMP(copy)	<b>57.09</b>	<b>43.0</b>	40.3

Table 1: Multi-BLEU, METEOR and TER on WebNLG 2017 challenge test dataset (seen entities).

Model	multi-BLEU $\uparrow$	METEOR $\uparrow$	TER $\downarrow$
BiLSTM	51.23	36.6	45.8
bi-GMP(copy)	55.99	41.0	41.8
bi-GCN(2L)	56.06	40.5	41.9
bi-GCN(3L)	55.86	40.3	41.8
bi-GCN(2L)+copy	56.74	41.1	40.4
bi-GCN(2L) + bi-GMP(copy)	<b>57.76</b>	<b>41.4</b>	<b>39.4</b>

Table 2: Multi-BLEU, METEOR and TER on WebNLG supplementary test dataset (seen entities).

## 6 Experiments

### 6.1 Datasets

We use two different WebNLG datasets<sup>1</sup> [Gardent *et al.*, 2017a] which are designed for the task of mapping RDF triples to text. Each example is a  $\langle \text{triples}, \text{text} \rangle$  pair. Here, one triple set may correspond to multiple reference texts. Each RDF triple is represented as  $\langle \text{subject}, \text{relationship}, \text{object} \rangle$ , where the *subject* and *object* are constants or entities. The first dataset is the WebNLG 2017 challenge dataset, consisting of 18102 training pairs, 2268 validation pairs, and 2495 test pairs in 10 categories (Astronaut, Building, Monument, University, SportsTeam, WrittenWork, etc.). The second supplementary dataset is extracted from an enriched version of the WebNLG 2017 challenge dataset. The enriched dataset consists of 31969 training pairs, 4030 validation pairs and 4222 test pairs. The second supplementary dataset contains 13867 training pairs, 1762 validation pairs, and 1727 test pairs, which does not overlap the first dataset. The second supplementary dataset belongs to 5 other categories (Athlete, Artist, MeanOfTransportation, CelestialBody, Politician).

### 6.2 Experimental Settings and Evaluation Metrics

We build a vocabulary list based on the training set, which is shared between the encoders and the decoder. For model hyperparameters, we set 300-dimension source and target word embeddings and 300-dimension hidden state for bi-GCN encoder, meta-paths encoder and decoder. We use Adam [Kingma and Ba, 2014] as the optimization method

<sup>1</sup><https://gitlab.com/shimorina>

with an initial learning rate 0.001 and learnable parameters are updated every 64 instances.

For our experiments, we adopt the standard evaluation metrics of the WebNLG challenge, including BLEU [Papineni *et al.*, 2002], METEOR [Denkowski and Lavie, 2011] and TER. The metric of BLEU suggested by the WebNLG challenge is multi-BLEU. For metrics BLEU and METEOR, the higher the better, while for metric TER, the lower the better.

### 6.3 Baselines

We compared our model against the following baselines:

**Sequential Model.** Sequential model contains an attention-based bidirectional LSTM encoder and an LSTM decoder. The RDF triples are transformed into a sequence. Moreover, the results of MELBOURNE and PKUWRITER are both sequential models, which are reported in [Gardent *et al.*, 2017b].

**DGCN Model.** We rerun the code of [Marcheggiani and Perez-Beltrachini, 2018] to get the results of Deep Graph Convolutional Encoder (DGCN).

**Bi-GMP Model.** Bi-GMP model consists of a meta-paths encoder calculating the representation of each token in the meta-paths from two directions and a one-layer LSTM decoder. Copy mechanism [Gu *et al.*, 2016] is incorporated into the model to improve the performance.

**Bi-GCN Model.** Bi-GCN model consists of a bi-GCN encoder (GCN with two layers) and a one-layer LSTM decoder. We exploit bi-GCN with- and without- copy mechanism.

### 6.4 Experimental Results

As shown in Tables 1 and 2, our proposed model consistently outperforms other baselines on all three evaluation metrics. For instance, our full model achieves about 1.0 multi-BLEU points higher than those of the other baselines on two WebNLG test datasets. This is because our full model could better capture the global and local graph structure of the RDF triples. In addition, Table 3 shows our proposed model achieve higher scores on BLEU-1, BLEU-2, BLEU-3 and BLEU-4 compared to other baselines, indicating that our full model can better encode multi-perspective information. Our code is publicly available for research purpose.<sup>2</sup>

### 6.5 Ablation Study

As shown in Table 1 and Table 2, there are three key factors in our proposed model that may affect the quality of generated text. The first two are the bi-GCN and bi-GMP encoders. Experimental results shows the model combining graph-augmented structural neural encoders performs better than the models with one single graph-based encoder. This result is expected, since it is difficult for one single graph-based encoder to fully encode both global and local structure information completely. The third factor is copy mechanism. We only apply copy mechanism on the bi-GMP encoder in our full model. Interestingly, we found that when applying copy mechanism on both the bi-GMP encoder and the bi-GCN encoder, it actually leads to worse performance compared with our full model.

<sup>2</sup><https://github.com/Nicoleqwert/RDF-to-Text>.



Model \ Dataset/Metric	WebNLG 2017 Challenge				WebNLG Supplementary			
	BLEU <sub>1</sub>	BLEU <sub>2</sub>	BLEU <sub>3</sub>	BLEU <sub>4</sub>	BLEU <sub>1</sub>	BLEU <sub>2</sub>	BLEU <sub>3</sub>	BLEU <sub>4</sub>
BiLSTM	81.6	58.9	42.7	31.3	76.4	58.1	44.8	34.7
Meta-paths(copy)	83.0	63.5	48.4	37.2	81.4	63.1	49.3	38.8
GCN(2L)	84.6	64.0	48.4	37.1	81.9	63.6	49.4	38.4
GCN(2L)+copy	<b>85.5</b>	<b>65.6</b>	50.1	38.7	82.9	64.1	49.9	39.1
GCN(2L) + Meta-paths(copy)	84.3	65.0	<b>50.1</b>	<b>38.8</b>	<b>83.3</b>	<b>65.6</b>	<b>51.7</b>	<b>40.9</b>

 Table 3: Detailed BLEU scores of the baseline models we adopt and our proposed model for BLEU<sub>1</sub>, BLEU<sub>2</sub>, BLEU<sub>3</sub> and BLEU<sub>4</sub>.

RDF Triples	(Denmark, leaderName, Lars Løkke Rasmussen), (European University Association, headquarters, Brussels), (School of Business and Social Sciences at the Aarhus University, country, Denmark), (Denmark, leaderTitle, Monarchy of Denmark), (School of Business and Social Sciences at the Aarhus University, affiliation, European University Association), (Denmark, religion, Church of Denmark)
Reference Output	The school of business and social sciences at the Aarhus University in Denmark is affiliated with the European University Association , which has its hq in Brussels . Denmark has a monarch ; its religion is the church of Denmark and its leader is Lars Løkke Rasmussen .
bi-GCN	Lars Løkke Rasmussen is the leader of Denmark which is led by the Monarchy of Denmark . The country is the location of the European University Association which has its headquarters in Brussels . The school of business and social sciences at the Aarhus University is located in the country .
bi-GMP	The school of business and social sciences at the Aarhus University is located in Brussels , Denmark . The school is affiliated with the European University Association and <b>its religion is the Monarchy of Denmark</b> . The leader of Denmark is Lars Løkke Rasmussen and <b>the religion is the Monarchy of Denmark</b> .
Our Model	Lars Løkke Rasmussen is the leader of Denmark . The country is the location of the school of business and social sciences at the Aarhus University which is affiliated with the European University Association which has its headquarters in Brussels . School of business and social sciences at the Aarhus university is located in Denmark .

Table 4: Sample outputs of different models. The bold tokens are error outputs.

Methods	Grammar	Informativity	Conciseness
Reference	4.46	4.82	4.65
bi-GMP	4.26	4.09	4.27
bi-GCN	4.21	3.98	4.44
Our Full Model	4.55	4.24	4.57

Table 5: Human evaluation of WebNLG 2017 Challenge test dataset. The higher the score is, the better the performance is.

## 6.6 Case Study

We further manually inspect the outputs of different models and conduct case studies for better understanding the model performances. As shown in Table 4, we find that the models involving bi-GCN encoder perform better on covering correct relationships between entities. This result is expected since the bi-GCN encoder pays more attention to the local structure information and can predict the intra-triple relationship more accurately and effectively. Meanwhile, the bi-GMP encoder mainly focuses on the inter-triple relationships, which helps model long-range dependency with more context.

## 6.7 Human Evaluation

To further evaluate the quality of these generated text examples, we presented a number of original RDF triples and corresponding generated text pairs to three human evaluators. Three evaluators are shown 200 outputs, a quarter references, a quarter generated by bi-GCN model, a quarter generated by bi-GMP model and a quarter generated by full model. The evaluators were asked to evaluate the generated texts from

three perspectives (each perspective scale from 1 to 5): I) *Grammatical* rates each text sample with respect to the coherence, no redundancy, and no grammatical errors; II) *Informativity (Global)* scores based on how well the information in input triples are covered; III) *Conciseness (Local)* evaluates whether the entities and corresponding relationships appearing in the generated sentence are accurate. We averaged the results from three evaluators for all three tasks. Table 5 indicates that text examples generated by our proposed model are indeed improved on the respects of informativity (global information) and conciseness (local information).

## 7 Conclusion

In this paper, we propose a novel approach via exploiting graph-augmented structural neural encoders for RDF-to-text generation. Our approach jointly learns structure information locally and globally via the combination of a bidirectional graph encoder and a bidirectional graph-based meta-paths encoder to learn intra-triple and inter-triple relationships. The experimental results show that our proposed model outperforms the state-of-the-art baselines. One of the future works is to further extend our proposed method to develop a knowledge graph question answering system.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.61402191), and the Thirteen Five-year Research Planning Project of National Language Committee (No.WT135-11).

## References

- [Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [Banik *et al.*, 2012] Eva Banik, Claire Gardent, Donia Scott, Nikhil Dinesh, and Fennie Liang. Kbgcn: text generation from knowledge bases as a new shared task. In *Proceedings of the Seventh International Natural Language Generation Conference*, pages 141–145, 2012.
- [Barzilay and Lapata, 2005] Regina Barzilay and Mirella Lapata. Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 331–338, 2005.
- [Bontcheva and Wilks, 2004] Kalina Bontcheva and Yorick Wilks. Automatic report generation from ontologies: the miakt approach. In *International conference on application of natural language to information systems*, pages 324–335. Springer, 2004.
- [Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.
- [Chen *et al.*, 2020] Yu Chen, Lingfei Wu, and Mohammed J Zaki. Reinforcement learning based graph-to-sequence model for natural question generation. In *ICLR*, 2020.
- [Deemter *et al.*, 2005] Kees Van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24, 2005.
- [Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.
- [Denkowski and Lavie, 2011] Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91, 2011.
- [Gao *et al.*, 2019] Yuyang Gao, Lingfei Wu, Houman Homayoun, and Liang Zhao. Dyngraph2seq: Dynamic-graph-to-sequence interpretable learning for health stage prediction in online health forums. In *ICDM*, 2019.
- [Gardent *et al.*, 2017a] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, August 2017.
- [Gardent *et al.*, 2017b] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, 2017.
- [Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.
- [Hao *et al.*, 2017] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 221–231, 2017.
- [Jagfeld *et al.*, 2018] Glorianna Jagfeld, Sabrina Jenne, and Ngoc Thang Vu. Sequence-to-sequence models for data-to-text natural language generation: Word-vs. character-based processing and output diversity. *arXiv preprint arXiv:1810.04864*, 2018.
- [Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [Liu *et al.*, 2018] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.
- [Marcheggiani and Perez-Beltrachini, 2018] Diego Marcheggiani and Laura Perez-Beltrachini. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9, 2018.
- [Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.
- [Moryossef *et al.*, 2019] Amit Moryossef, Yoav Goldberg, and Ido Dagan. Step-by-step: Separating planning from realization in neural data-to-text generation. *arXiv preprint arXiv:1904.03396*, 2019.
- [Papineni *et al.*, 2002] 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 on association for computational linguistics*, pages 311–318, 2002.
- [Pouriyeh *et al.*, 2017] Seyedamin Pouriyeh, Mehdi Allahyari, Krzysztof Kochut, Gong Cheng, and Hamid Reza Arabnia. Es-lda: Entity summarization using knowledge-based topic modeling. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 316–325, 2017.
- [See *et al.*, 2017] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.
- [Song *et al.*, 2018] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*, 2018.
- [Trisedya *et al.*, 2018] Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. Gtr- lstm: A triple encoder for sentence generation from rdf data. In *ACL*, volume 1, pages 1627–1637, 2018.
- [Xu *et al.*, 2018] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, Michael Witbrock, and Vadim Sheinin. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*, 2018.
- [Zhu *et al.*, 2019] Yaoming Zhu, Juncheng Wan, Zhiming Zhou, Liheng Chen, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. Triple-to-text: Converting rdf triples into high-quality natural languages via optimizing an inverse kl divergence. *arXiv preprint arXiv:1906.01965*, 2019.