Few-shot Knowledge Graph-to-Text Generation with Pretrained Language Models

Junyi Li[1,3], **Tianyi Tang**[1], **Wayne Xin Zhao**[1,3,5*], **Zhicheng Wei**[4], **Nicholas Jing Yuan**[4] and **Ji-Rong Wen**[1,2,3]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Huawei Cloud
[5]Beijing Academy of Artificial Intelligence, Beijing, 100084, China
{lijunyi,jrwen,steven_tang}@ruc.edu.cn
{batmanfly,nicholas.jing.yuan}@gmail.com weizhicheng1@huawei.com

Abstract

This paper studies how to automatically generate a natural language text that describes the facts in knowledge graph (KG). Considering the few-shot setting, we leverage the excellent capacities of pretrained language models (PLMs) in language understanding and generation. We make three major technical contributions, namely representation alignment for bridging the semantic gap between KG encodings and PLMs, relation-biased KG linearization for deriving better input representations, and multi-task learning for learning the correspondence between KG and text. Extensive experiments on three benchmark datasets have demonstrated the effectiveness of our model on KG-to-text generation task. In particular, our model outperforms all comparison methods on both fully-supervised and few-shot settings. Our code and datasets are available at https://github.com/RUCAIBox/Few-Shot-KG2Text.

1 Introduction

Knowledge graphs (KGs), such as Wikidata and DBpedia, are essential for many natural language processing (NLP) applications (Ji et al., 2020). To understand the structured information in KG, the task of KG-to-text generation has been proposed to automatically generate a descriptive text for a given knowledge graph (Koncel-Kedziorski et al., 2019; Ribeiro et al., 2020a). Figure 1 illustrates a KG with the corresponding descriptive text, in which the nodes (*e.g., Stan Lee* and *Iron Man*) represent entities and the edges (*e.g., creator* and *alias*) describe the relations between connected entities.

In recent years, with the help of crowdsourcing platforms and information extraction (IE) systems, large-scale labelled pairs of KG and its descriptive text have been created, such as WikiBio (Lebret et al., 2016) and WebNLG Challenge (Gardent et al., 2017). Based on these datasets, data-driven models have shown impressive capabilities to produce informative and fluent text for a given KG (Logan et al., 2019; Moryossef et al., 2019). However, due to the great expense in annotation process, it is not always feasible to generate large-scale labelled datasets for a variety of domains in practice. Motivated by this, we propose to study the task of *few-shot KG-to-text generation* that aims to produce satisfactory output text given only a handful of (several hundred) labelled instances.

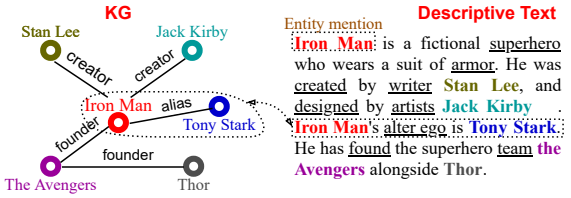


Figure 1: A knowledge graph (subgraph) with its descriptive text. The underlined words represent the context keywords about entities.

To fulfil this task, we need to fully understand the complicated semantic relations between entities from various domains, which is challenging with limited labelled data. Our solution is inspired by the excellent few-shot capabilities of pretrained language models (PLMs) on language understanding and generation tasks (Brown et al., 2020; Chen et al., 2020; Li et al., 2021a). Pretrained on the large-scale corpora, PLMs encode vast amounts of world knowledge into their parameters (Li et al., 2021b), which is potentially beneficial to understand and describe the KG facts in our task.

However, applying PLMs to few-shot KG-to-text generation still faces two challenges. First, PLMs are usually pretrained on natural language text, while the KG inputs for our task are structured graphs. This semantic gap makes it difficult to effectively inject KG representations into PLMs

*Corresponding author

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1558–1568

especially with limited labelled instances. Second, unlike many other text generation tasks, KG-to-text generation requires faithful generation based on the understanding of KG facts. It needs to learn an accurate semantic correspondence between input KG and output text, which will be more difficult in few-shot settings.

To address the above issues, in this paper, we propose a few-shot KG-to-text generation model based on PLMs. There are three major technical contributions in our model. First, in order to bridge the semantic gap, we enforce the representation alignment by learning the correspondence between KG representations (encoded by graph neural networks) and PLM-based entity representations. Second, to feed KG into PLMs, we propose a relation-biased breadth-first search (RBFS) strategy to linearize KG into a well-planned entity sequence. Finally, we jointly train the primary text generation task and an auxiliary KG reconstruction task under the framework of multi-task learning. This step further enhances the semantic correspondence between input KG and output text, based on which our model can generate faithful text about KG.

To the best of our knowledge, we are the first study to investigate PLMs for few-shot KG-to-text generation. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our few-shot KG-to-text generation model.

2 Related Work

In this work, we mainly focus on generating text from knowledge graphs using PLMs.

KG-to-Text Generation. Early works mainly centered around statistical methods, applying grammar rules to generate text (Konstas and Lapata, 2013; Flanigan et al., 2016). Recently, neural based approaches have been proposed to generate text from linearized KG triples (Gardent et al., 2017), however, unable to model structural information about KG. Many works explored how to encode the graph structure using Graph Neural Networks (GNNs) or Transformers explicitly. Koncel-Kedziorski et al. (2019) leveraged a graph Transformer encoder to compute node representations by attending over local neighborhoods via self-attention. In contrast, Ribeiro et al. (2020a) focused on combining global and local message passing mechanisms based on GNNs, capturing complementary graph contexts. Guo et al. (2020) presented an unsupervised training method that can iteratively back translate between the text and graph data. Different from them, we explore how to utilize large PLMs for few-shot KG-to-text generation.

Pretrained Language Model. Recent years have witnessed prominent achievement of PLMs in NLP tasks (Devlin et al., 2019; Radford et al., 2019). Pretrained on massive corpora, pretrained models showcase unprecedented generalization ability to solve related downstream tasks (Li et al., 2021b). However, most of existing PLMs were conditioned on text data (Radford et al., 2019; Lewis et al., 2020), lacking consideration of structured data input. Ribeiro et al. (2020b) proposed to utilize PLMs for KG-to-text generation by randomly linearizing graph into a sequence of triples. While, these methods do not explicitly model the structural relations of KG, which is critical for generating faithful text. Our work aims to consider the KG structure and bridge the semantic gap between KG encodings and PLMs.

3 Problem Formulation

KG-to-text generation (Ribeiro et al., 2020a) aims to automatically generate a natural language text that describes the facts in KG.

Formally, the input KG consists of a set of triples, denoted as $\mathcal{G} = \{\langle e, r, e' \rangle | e, e' \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and relation set, respectively. A triple $\langle e, r, e' \rangle$ denotes the fact that relation r exists between head entity e and tail entity e'. Note that the input KG is a small and compact subgraph extracted from large-scale knowledge graphs (*e.g.,* DBpedia). Following Koncel-Kedziorski et al. (2019), a text describing the input KG is usually available in this task. Let $\mathcal{V}$ denote the vocabulary. The target is to generate a natural language text $\mathcal{Y} = \langle w_1, ..., w_j, ..., w_T \rangle (w_j \in \mathcal{V})$ that represents the correct and concise semantics of entities and their relations in the given knowledge graph. The text contains a set of entity mentions $\mathcal{M} = \{m_e | m_e = \langle e, s_e, o_e \rangle, e \in \mathcal{E}\}$, where e is the target entity, s_e and o_e are the start and end indices of this mention in text $\mathcal{Y}$, respectively. In other words, $\langle w_{s_e}, ..., w_{o_e} \rangle$ specially corresponds to entity e. For entities with multiple mentions in text, we only keep the first mention of each entity in $\mathcal{M}$. By replacing each word of mentions with the token "[MASK]", we can obtain a masked text, denoted as $\mathcal{Y}_{[mask]}$, which is also taken as input for representation alignment in Section 4.1.

In practice, it is difficult to collect massive pairs

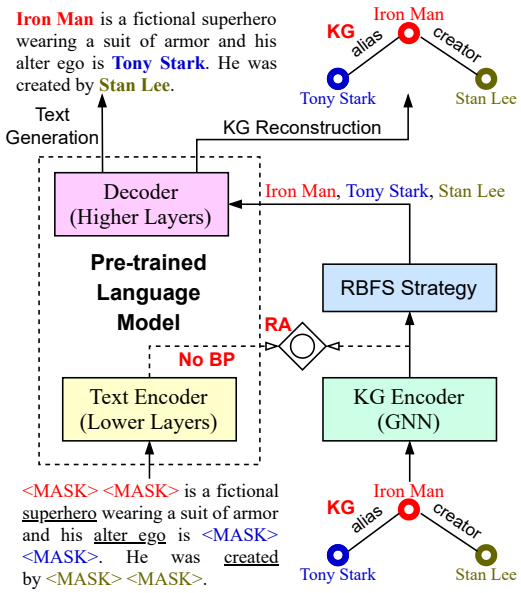


Figure 2: Overview of our proposed model. "RA" and "BP" denote representation alignment and back propagation, respectively. We organize the PLM into lower layers and higher layers. The former provides PLM-based entity representations for alignment with KG encodings, and the latter acts as a decoder for generating text and reconstructing KG facts. After representation alignment, KG embeddings can be directly fed into the higher layers of PLMs for generating text.

of KG and its descriptive text for training. In this paper, we study the task of *few-shot KG-to-text generation* with a handful of training instances (*e.g.,* 200 instances) based on a given PLM (*e.g.,* GPT-2).

4 Approach

For our task, two major challenges are how to learn effective input representations and capture the semantic correspondence between KG and text. To address the two challenges, we propose three major technical contributions, namely representation alignment between KG encodings and PLMs, relation-biased BFS strategy for KG linearization, and multi-task learning with KG reconstruction. Figure 2 presents an illustrative overview of our model. Next we will describe each part in detail.

4.1 Representation Alignment

Unlike previous works (Ribeiro et al., 2020b; Yang et al., 2020) that directly transform KG into text sequence, we employ graph neural network (GNN) as knowledge graph encoder to explicitly encode entity relations in KG. Based on the input KG, GNN would produce a set of entity embeddings, which can be regarded as the input word embeddings of PLM for generating text. However, the GNN-based entity embeddings and the PLM-based word (entity) embeddings come from two distinct semantic spaces. To bridge such a semantic gap, we propose a representation alignment method to align the GNN-based and PLM-based entity embeddings in different semantic spaces.

KG Encoder. The GNN-based KG encoder aims to generate entity embeddings for KG. Let $\boldsymbol{v}_e \in \mathbb{R}^{d_E}$ denote the entity embedding for a general entity e in KG, where d_E is the embedding size. In our work, the entity embeddings are shared across different KGs and initialized with pretrained KG embeddings (Yang et al., 2015). We apply R-GCN (Schlichtkrull et al., 2018) to generate entity embeddings by leveraging multi-relational information in KG. Then, the embedding of entity e at the $l+1$-th layer of R-GCN can be computed as:

$$\boldsymbol{v}_e^{(l+1)} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{e'\in\mathcal{N}_e^r} \boldsymbol{W}_r^{(l)}\boldsymbol{v}_{e'}^{(l)} + \boldsymbol{W}_0^{(l)}\boldsymbol{v}_e^{(l)}\Big), \quad (1)$$

where $\boldsymbol{W}_0^{(l)}$ and $\boldsymbol{W}_r^{(l)}$ are trainable matrices, and $\mathcal{N}_e^r = \{e' | \langle e, r, e'\rangle, \langle e', r, e\rangle \in \mathcal{G}\}$ denotes the set of neighbors of entity e under relation r. Finally, after stacking L times, the output entity embedding $\boldsymbol{v}_e^{(L)}$ from the last R-GCN layer is used as the final entity embedding $\tilde{\boldsymbol{v}}_e$.

Note that, we represent an entity as a set of nodes. For instance, the entity *Iron Man* in Figure 1 will be represented by two nodes: one for the token *Iron* and the other for the token *Man*. This would enhance the generalization ability of KG encoder on unseen entities, since it learns entity embeddings at the token level.

Text Encoder. To obtain the PLM-based entity embeddings, we feed the masked text $\mathcal{Y}_{[mask]}$ into the text encoder, *i.e.,* the lower layers of PLM. As shown in Figure 1, compared with short entity mentions, the masked text contains rich context information about entities. Therefore, similar to masked language model (Devlin et al., 2019), the embeddings of masked text can be computed as:

$$\langle \hat{\boldsymbol{v}}_{w_1}, ..., \hat{\boldsymbol{v}}_{w_T} \rangle = \text{Lower-Layers}(\mathcal{Y}_{[mask]}), \quad (2)$$

where the entity mention m_e corresponds to the embedding sequence $\langle \hat{\boldsymbol{v}}_{w_{s_e}}, ..., \hat{\boldsymbol{v}}_{w_{o_e}} \rangle$ and the PLM-based entity embedding $\hat{\boldsymbol{v}}_e$ can be computed by an average pooling over this embedding sequence.

To bridge the semantic gap, we model the representation alignment by minimizing the Euclidean distance in semantic space between the GNN-based and PLM-based entity embeddings as:

$$\mathcal{L}_{RA} = \sum_{e \in \mathcal{E}} \|\tilde{\boldsymbol{v}}_e - \hat{\boldsymbol{v}}_e\|_2, \quad (3)$$

where $\tilde{\boldsymbol{v}}_e$ and $\hat{\boldsymbol{v}}_e$ are GNN-based and PLM-based entity embeddings, respectively.

With representation alignment, the GNN-based entity embeddings can be aligned with the PLM-based entity embeddings in semantic space, which enables us to effectively inject KG representations into PLM for improving generation quality.

4.2 Knowledge Graph Linearization

To feed the KG into decoder (*i.e.,* the higher layers of PLM), we need to linearize KG into an entity sequence. Previous work (Yang et al., 2020; Ribeiro et al., 2020b) usually relies on random or pre-defined rules, which is not flexible to model KG structures. Here, we propose to utilize breadth-first search (BFS) strategy to traverse KG. BFS, a graph traversal algorithm, starts at the root node and explores all the nodes at the present layer before moving on to the nodes at the next depth layer[1]. Here, we assume that nodes at the same layer potentially express relevant semantics and should be placed in close positions of the entity sequence.

Furthermore, we observe that some relations are often lexicalized before others, *e.g.,* the nationality of a person often precedes the birthplace in descriptive text. Considering such relation priority, in this paper, we propose a *relation-biased breadth first search (RBFS)* strategy to traverse and linearize KG into entity sequence. Specifically, we first compute RBFS weights $\alpha_{e'}$ for each entity e' based on their relations as:

$$\alpha_{e'} = \sigma(\tilde{\boldsymbol{v}}_e^\top \boldsymbol{W}_r^{(L)} \tilde{\boldsymbol{v}}_{e'}), \ \langle e, r, e' \rangle \in \mathcal{G}, \quad (4)$$

where $\boldsymbol{W}_r^{(L)}$ is a relation matrix from Eq. 1. Then, for two sibling entities e' and e'' at the same layer, we traverse e' before e'' if $\alpha_{e'}$ is greater than $\alpha_{e''}$, and vice versa. Finally, through RBFS, we can obtain a linearized entity sequence taken as input of the decoder for text generation.

4.3 KG-enhanced Multi-task Learning

After obtaining the linearized entity sequence, we next take it as input and perform text generation. Different from other text generation tasks, KG-to-text generation aims to generate text reflecting the concise facts in KG. Inspired by Liu et al. (2019), we incorporate an auxiliary KG reconstruction task to reconstruct the facts in KG for learning the semantic correspondence between text and KG.

Text Generation. The text generation task is performed upon the higher layers of PLM. The objective is to maximize the likelihood of the reference text, which is equivalent to minimize the negative log-likelihood as:

$$\mathcal{L}_{LM} = -\sum_{j=1}^{T} \log p_{gen}(w_j | w_1, ..., w_{j-1}; \mathcal{G}), \quad (5)$$

where p_{gen} is the generative probability from PLM. Besides, in KG-to-text generation, some tokens in descriptive text correspond to KG entities shown in Figure 1. The ability to copy entities from KG would enrich the generated text content, which can be achieved by the pointer generator (See et al., 2017). By feeding the hidden states of PLM and the token embedding, the copy probability p^j_{copy} of the j-th token w_j can be computed as:

$$p^j_{copy} = \sigma(\boldsymbol{W}_1 \boldsymbol{s}_j + \boldsymbol{W}_2 \boldsymbol{v}_{w_j} + b_{copy}), \quad (6)$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and b_{copy} are trainable parameters, $\boldsymbol{v}_{w_j}$ is the embedding of token w_j, and $\boldsymbol{s}_j$ is the j-th hidden state from the top layer of PLM. Then, we explicitly "teach" our model how to switch between generation and copy via the copy loss as:

$$\mathcal{L}_{PG} = \sum_{w_j} p^j_{copy} + \sum_{w_k} (1 - p^k_{copy}). \quad (7)$$

Our intuition is aimed at minimizing the copy probability p^j_{copy} of token w_j (generated from vocabulary) and maximizing the copy probability p^k_{copy} of token w_k (copied from KG entities).

KG Reconstruction. Following Song et al. (2020), we formalize the KG reconstruction task as predicting the relations between any two entities. In detail, given a head entity e and a tail entity e' in generated text, we can obtain the hidden states of their mentions from the top layer of decoder, *i.e.,* $\langle \boldsymbol{s}_{s_e}, ..., \boldsymbol{s}_{o_e} \rangle$ and $\langle \boldsymbol{s}_{s_{e'}}, ..., \boldsymbol{s}_{o_{e'}} \rangle$. Then, the entity hidden states $\boldsymbol{h}_e$ and $\boldsymbol{t}_{e'}$ can be computed by an average pooling over their mention hidden states. The probability for a relation r is calculated as:

$$p(r|e, e') = \text{softmax}(\boldsymbol{W}_3[\boldsymbol{h}_e; \boldsymbol{t}_{e'}; \boldsymbol{h}_e \odot \boldsymbol{t}_{e'}] + \boldsymbol{b}_2), \quad (8)$$

[1] https://en.wikipedia.org/wiki/Breadth-first_search

where $\boldsymbol{W}_3$ and $\boldsymbol{b}_2$ are trainable parameters. The loss for reconstructing KG is also defined as the negative log-likelihood of all target triples in KG:

$$\mathcal{L}_{GR} = -\sum_{\langle e,r,e'\rangle \in \mathcal{G}} \log p(r|e,e'). \quad (9)$$

By incorporating the KG reconstruction task, our model is able to capture the semantic correspondence between input KG and output text, which further improves generating faithful text.

Finally, the total training loss consists of text generation loss $\mathcal{L}_{LM}$ (Eq. 5), copy loss $\mathcal{L}_{PG}$ (Eq. 7), representation alignment loss $\mathcal{L}_{RA}$ (Eq. 3) and KG reconstruction loss $\mathcal{L}_{GR}$ (Eq. 9) as:

$$\mathcal{L}_{total} = \mathcal{L}_{LM} + \lambda_1 \mathcal{L}_{PG} + \lambda_2 \mathcal{L}_{RA} + \lambda_3 \mathcal{L}_{GR}, \quad (10)$$

where λ_1, λ_2 and λ_3 are combination coefficients.

4.4 Discussion and Learning

In this part, we present the model discussion and the model optimization.

Few-shot Learning. In few-shot KG-to-text generation, the key lies in how to bridge the semantic gap between KG and PLMs with limited dataset. To achieve this goal, we first utilize representation alignment in Section 4.1 to align the semantic space between KG encodings and PLMs, and then introduce a KG reconstruction task in Section 4.3 to further learn the semantic correspondence between input KG and output text. Besides, we observe that KG entities are often multi-word expressions. To deal with unseen entities in few-shot learning, we employ the Byte Pair Encoding (BPE) (Sennrich et al., 2016) and sub-word vocabulary (Radford et al., 2019) to split entity words into smaller semantic units. Our work is also empowered by the excellent few-shot capacities of PLMs with vast amounts of world knowledge learned from large-scale corpora.

Optimization. For PLM, we employ BART-Large model (Lewis et al., 2020). Specially, we adopt the first 6 layers of BART encoder as the lower layers, and the remaining 6 layers of BART encoder and BART decoder as the higher layers. Note that, the target text and text encoder will not be used at test time. In particular, the target text is just used at training time and encoded as PLM-based entity embeddings for representation alignment, while the alignment is not needed at test time. We optimize all parameters according to the total loss in Eq. 10 with the OpenAI AdamW optimizer (Loshchilov and Hutter, 2019). The learning rate, batch size, R-GCN layers and embedding size are set to 1e-5, 20, 2 and 1024, respectively. The weights λ_1, λ_2 and λ_3 in Eq. 10 are set to 0.7, 0.5 and 0.5, respectively, according to performance on validation set. During inference, we apply the beam search method with a beam size of 8.

Dataset	#Train	#Valid	#Test	#Relations
AGENDA	29,720	1,000	10,000	42
WebNLG	7,362	1,389	5,427	107
GenWiki	48,020	1,000	10,000	250

Table 1: Statistics of three datasets.

5 Experiments

In this section, we first set up the experiments, and then report the results and analysis.

5.1 Experimental Setup

Datasets. To evaluate our model on few-shot KG-to-text generation, we conduct experiments on three benchmarks, including AGENDA (Koncel-Kedziorski et al., 2019), WebNLG (Gardent et al., 2017) and GenWiki Fine (Jin et al., 2020). We adopt three large domains (*i.e., Airport*, *Building* and *Food*) for WebNLG and two large domains (*i.e., Sports* and *Games*) for GenWiki. Table 1 shows the statistics for each dataset. Each instance of these datasets contains a knowledge graph in the form of triples and a target text describing the graph. The three datasets have originally provided the alignment records from entity mentions to KG entities. Take an example from WebNLG dataset "AGENT-1 is located in PATIENT-1": the entity mention is tagged as "AGENT-1" and the tag "AGENT-1" maps to the entity "11th_Mississippi_Infantry_Monument" in KG. If such alignments are not available, we can utilize entity linking tools (*e.g.,* NER packages) for preprocessing.

Baselines. We make a comparison against five KG-to-text generation models:

• *GraphWriter* (Koncel-Kedziorski et al., 2019) introduces a graph transformer encoder and a sequence decoder for generating text based on KG.

• *CGE-LW* (Ribeiro et al., 2020a) proposes a graph-to-text model by combining both global and local node aggregation strategies.

Datasets	AGENDA				WEBNLG				GENWIKI FINE			
#Metrics	B-4	R-L	CIDEr	Chrf	B-4	R-L	CIDEr	Chrf	B-4	R-L	CIDEr	Chrf
GraphWriter	15.30	22.03	0.24	38.33	45.84	60.62	3.14	55.53	29.73	55.46	2.68	46.87
CGE-LW	18.01	25.62	0.33	46.69	48.60	62.52	3.85	58.66	30.67	56.37	3.20	47.79
CycleGT	20.16	25.77	0.69	48.26	50.20	68.30	3.81	68.91	38.57	59.37	3.50	62.46
BART-base	22.01	26.44	0.90	48.02	49.81	63.10	3.45	67.65	48.20	59.21	4.02	65.80
BART-large	23.65	28.76	1.15	50.44	52.49	65.61	3.50	72.00	**50.70**	61.90	4.51	68.15
T5-base	20.59	29.41	0.81	48.15	48.86	65.57	3.99	66.08	45.72	58.28	3.74	65.68
T5-large	22.15	30.68	0.87	48.88	58.78	68.22	4.10	74.40	47.11	60.64	3.74	**68.47**
Ours	**25.15**	**35.12**	**3.23**	**55.89**	**61.88**	**75.74**	**6.03**	**79.10**	48.46	**65.65**	**5.19**	64.00

Table 2: Performance comparisons of different methods for fully-supervised KG-to-text generation under three domains. B-n and R-n are short for BLEU-n and ROUGE-n. **Bold** and underline fonts denote the best and the second best methods (the same as below).

Datasets	AGENDA				WEBNLG				GENWIKI FINE			
#Instances	50	100	200	500	50	100	200	500	50	100	200	500
BART-large	5.71	6.15	7.59	10.71	9.05	15.70	19.38	27.91	9.14	13.38	15.39	24.14
T5-large	2.69	2.73	4.65	7.52	7.18	14.52	16.88	21.68	6.30	6.36	10.37	17.72
Ours	**6.22**	**9.40**	**10.21**	**17.93**	**10.60**	**17.46**	**20.00**	**31.79**	**10.75**	**14.44**	**16.84**	**28.89**

Table 3: BLEU-4 results of different methods for few-shot KG-to-text generation under three domains. To mitigate the randomized effects of samples, we report the average results over five training runs (the same as below).

Datasets	AGENDA				WEBNLG				GENWIKI FINE			
#Instances	50	100	200	500	50	100	200	500	50	100	200	500
BART-large	14.33	15.28	16.94	20.70	22.57	26.21	30.68	49.34	26.59	29.60	34.56	47.50
T5-large	14.11	14.17	15.88	21.72	20.80	22.71	24.18	38.36	21.02	21.36	20.07	35.72
Ours	**15.10**	**16.65**	**18.88**	**25.72**	**24.80**	**28.38**	**33.12**	**55.13**	**28.02**	**31.36**	**38.07**	**50.72**

Table 4: ROUGE-L results of different methods for few-shot KG-to-text generation under three domains.

• *CycleGT* (Guo et al., 2020) jointly learns two dual tasks (graph-to-text generation and text-to-graph relation classification) via cycle training.

• *BART-Base/Large* (Ribeiro et al., 2020b) linearizes the KG into sequence and applies BART-Base/Large (Lewis et al., 2020) to generate text.

• *T5-Base/Large* (Ribeiro et al., 2020b) linearizes KG into a triple sequence and employs T5-Base/Large (Raffel et al., 2020) to generate text.

Among these baselines, *GraphWriter* and *CGE-LW* are GNN-based generation models; *CycleGT* is an unsupervised model using cycle training; *GPT2-Base/Large* and *BART-Base/Large* are the most relevant comparisons, which also employ PLMs in KG-to-text generation. These baselines were trained on the whole training dataset, *i.e.,* all KG-text pairs. Following previous few-shot work (Chen et al., 2020), we train our model on different few-shot settings with training dataset size ranging from 50, 100, 200 to 500. All the comparison methods are optimized based on validation performance. In our model, the entity embeddings of GNN are initialized with pretrained KG embeddings and the GNN weights are transferred from CGE-LW. We also pretrain GNN weights based on the large-scale KG, *i.e.,* Wikipedia. Based on the pretrained entity embeddings and weights, we continue to train our model.

Evaluation Metrics. For performance comparison, we adopt five automatic evaluation metrics widely used by previous graph-to-text work (Guo et al., 2020), *i.e.,* BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and CHRF++ (Popovic, 2015). Specifically, BLEU-n and ROUGE-n compute the ratios of overlapping n-grams between generated and real text, CIDEr computes the TF-IDF weights for each n-gram in generated/real text, and CHRF++ computes F-score averaged on both character- and word-level n-grams.

Models	B-4	R-L	CIDEr	Chrf
Ours	**31.79**	**55.13**	**3.94**	**57.38**
w/o RA	23.14	41.34	1.90	43.34
w/o GR	27.56	46.69	2.82	48.90
w/o PG	29.30	48.66	3.58	53.44

Table 5: Ablation analysis on WEBNLG dataset.

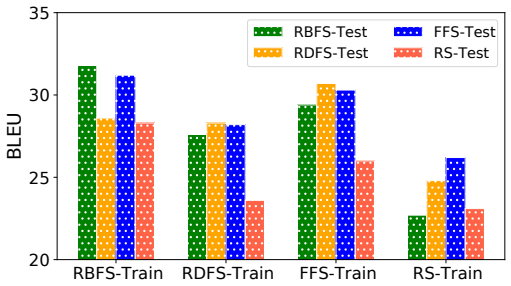


Figure 3: Linearization analysis on WEBNLG dataset.

Models	#Supp.↑	#Cont.↓	Naturalness↑
Gold	4.40	0.36	4.26
Ours	3.77	1.01	3.96
BART-Large	3.20	1.90	3.55
CEG-LW	2.87	2.13	2.56

Table 6: Human evaluation on WEBNLG dataset. Cohen's kappa coefficients for labelling three factors are as follows: 0.78, 0.71, and 0.75.

5.2 Main Results

Table 2, 3, and 4 present the fully-supervised and few-shot results of our model and other baselines, respectively.

First, by combining global and local entity context, CGE-LW performs better than GraphWriter. Furthermore, with two elaborate designed dual tasks, CycleGT becomes the best non-PLM baseline, outperforming GraphWriter and CGE-LW.

Second, as the most direct comparison with our model, BART-Base/Large and T5-Base/Large perform better than baselines by leveraging encoded semantics in PLMs, which reveals the feasibility of utilizing PLMs for KG-to-text generation.

Finally, we observe that our model achieves the best performance on both fully-supervised and few-shot settings. Large-scale PLMs can encode world knowledge by reading a large amount of text, making it easier to recover KG facts. Given only a handful of examples, the performances of baselines drop drastically, while the performance of our model only descents slightly. Furthermore, with only 500 labelled instances, our model improves over CGE-LW and CycleGT, and achieves the best performance in most cases. Compared to these PLM-based KG-to-text baselines, we adopt GNN to explicitly encode KG structure and representation alignment to bridge the semantic gap between PLM and GNN. This helps produce effective semantic representations for few-shot learning. Furthermore, we incorporate an auxiliary KG reconstruction task to learn semantic correspondence between input KGs and output text. These results indicate that our model can achieve more superior performance on KG-to-text generation task in a few-shot setting.

5.3 Detailed Analysis

Next, we conduct detailed analysis experiments on our model. We only report the test results on WEBNLG dataset with 500 training instances due to similar findings in other datasets.

Ablation Analysis. In our ablation study, we evaluate the effect of each loss $\mathcal{L}_{PG}$, $\mathcal{L}_{RA}$ and $\mathcal{L}_{GR}$ on the overall model performance. Here, we consider three variants:

• *w/o PG*: the variant removes the copy loss $\mathcal{L}_{PG}$.

• *w/o RA*: the variant removes the representation alignment loss $\mathcal{L}_{RA}$.

• *w/o GR*: the variant removes the KG reconstruction loss $\mathcal{L}_{GR}$.

As can be seen from Table 5, by removing any of the three losses, the BLEU/ROUGE/CIDEr performance drops compared to the complete model, especially removing $\mathcal{L}_{RA}$ and $\mathcal{L}_{GR}$. The proposed representation alignment bridges the semantic gap between PLM and GNN, which is helpful for adapting KG representations to PLM. The KG reconstruction task learns the correspondence between KG and text ensuring faithful generation about KG. We also observe a small performance drop by removing $\mathcal{L}_{PG}$. It is likely because PLM has learned some common phrase expressions about these KG facts from large-scale pretraining corpus.

KG Linearization Analysis. In Section 4.2, we propose a novel relation-biased BFS (RBFS) strategy to linearize the input KG into entity sequence. To verify the effectiveness of this strategy, we conduct linearization analysis by comparing RBFS with three traversal strategies, including relation-biased depth-first search (RDFS), forest fire search (FFS) and random search (RS). Specifically, RDFS combines both DFS and the relation factor similar

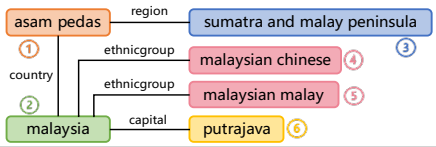
Real	Knowledge Graph		
	Reference	**asam pedas** is a food found in the region of **sumatra and malay peninsula** in **malaysia** , the capital of which is **putrajaya** , and whose ethnic groups include **malaysian malay** and **malaysian chinese** .	**athens international airport** serves the city **athens** in **greece** , **greek language** is spoken in greece and the leaders names in greece are **alexis tsipras** and **nikos voutsis** .
BART	Linearized KG	①③→①②→①⑥→②⑤→②④	①②→②④→②⑤→①③→②⑥
	Generated Text	**asam pedas** is a dish from **malaysia and sumatra** where the capital is **putrajava** . **malaysian malay** and **chinese** are ethnic groups in sumatra .	**athens** in **greece** is led by **alexis tsipras** and is served by **athens international airport** greece speaks **greek language** .
Ours	Linearized KG	①→③→②→⑥→⑤→④	①→③→②→⑥→⑤→④
	Generated Text	**asam pedas** comes from the region of **sumatra and malay peninsula** in **malaysia** , where the capital is **putrajava** , **malaysian malay** and **malaysian chinese** are ethnic groups .	**athens** is served by **athens international airport** in **greece** , which speaks **greek** textbflanguage . greece is led by **alexis tsipras** and **nikos voutsis** .

Table 7: Sample text generated by BART-Large baseline and our model from the *Food* and *Airport* domains of the WEBNLG benchmark. Since BART linearizes KG as triple sequence and an entity may involve in several triples, there are repeated entities used by BART (we omit the relations between entities). **Bold** and underlined words correspond to entity words and keywords.

to RBFS, where DFS starts at the root node and explores as far as possible along each branch before backtracking[2]; FFS is a randomized version of RBFS randomly exploring all the nodes at the same layer (Leskovec and Faloutsos, 2006); and RS randomly traverses all the nodes in the input KG. By re-training our model with the above three strategies, we report the comparison of BLEU results in Figure 3. It can be observed that, RBFS and FFS strategies achieve better results compared to the rest strategies. Nodes at the same layer tend to express more relevant semantics, thus searching by layer could produce more reasonable and coherent entity sequence especially considering the relations of entities as our RBFS strategy.

Human Evaluation. Following previous work in data-to-text (Chen et al., 2020), we conduct human evaluation on the generated text. We randomly sample 200 KG subgraphs along with corresponding generated text from CGE-LW, BART-Large and our model. In order to reduce the variance caused by human, three workers were asked to score the text with respect to two aspects: *Factual correctness* and *Language naturalness*. The first criterion evaluates how well the generated text correctly conveys information in the KG, by counting the number of facts in text supported by the KG (denoted as #Supp.) and contradicting with or missing from the KG (denoted as #Cont.). The second criterion evaluates whether the generated text is grammatically correct and fluent. The scoring mechanism adopts a 5-point Likert scale (Likert, 1932), ranging from 1-point ("very terrible") to 5-point ("very satisfying"). We further average the three scores from the three human judges over the 200 inputs. The results in Table 6 show that our model produces more fidelity and fluent texts than previous models. In our approach, the KG reconstruction task and pointer generator enhance the awareness of KG facts and alleviate producing incorrect facts. Also, with some learned common phrase expressions in PLMs, our model can generate natural text while keeping fidelity.

Qualitative Analysis. In this part, we present intuitive explanations why our model performs well. Table 7 presents two descriptions and the corresponding generated entity sequences and texts by BART-Large baseline and our model. As we can see, based on KG linearization, the generated texts by our model show reasonable and similar content sketch with real texts (*e.g., peninsula (region)→malaysia (country)→putrajava (capital)*).

[2]https://en.wikipedia.org/wiki/Depth-first_search

Besides, the baseline model incorrectly merges and generates unfaithful facts (*e.g., malaysia and sumatra*) or misses facts (*e.g., nikos voutsis*), while our model describes all the KG facts correctly. This improvement could be attributed to the KG reconstruction task, which enables our model to learn the correspondence between the input KG facts and output text. Finally, the entity words in our generated text are enriched and connected by meaningful keywords (*e.g.,* entity *greek language* and keyword *speaks*). The reason might be that, with the help of representation alignment, the GNN entity embeddings are aligned with the PLM word embeddings.

6 Conclusion

This paper presented a few-shot KG-to-text generation model based on PLMs. We make three important technical contributions, namely representation alignment for bridging the semantic gap between KG encodings and PLMs, relation-biased KG linearization for deriving better input KG representations, and multi-task learning for learning the correspondence between KG and text. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our few-shot KG-to-text generation model. As future work, we will consider adopting KG-enhanced PLMs (Zhang et al., 2019; Peters et al., 2019) for improving the task performance, which explicitly inject knowledge information into PLMs.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872369 and 61832017, Beijing Academy of Artificial Intelligence (BAAI) under Grant No. BAAI2020ZJ0301, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China under Grant No.18XNLG22 and 19XNQ047. Xin Zhao is the corresponding author.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 183–190. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime G. Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *NAACL HLT 2016, San Diego California, USA, June 12-17, 2016*, pages 731–739. The Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 124–133. Association for Computational Linguistics.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. *arXiv preprint arXiv:2006.04702*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *CoRR*, abs/2002.00388.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. Genwiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2398–2409. International Committee on Computational Linguistics.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph

transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2284–2293. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2013. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1503–1514. ACL.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213. The Association for Computational Linguistics.

Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 631–636. ACM.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Gaole He, Jinhao Jiang, Xiaoxuan Hu, Puzhao Xie, Zhipeng Chen, Zhuohao Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2021a. TextBox: A unified, modularized, and extensible framework for text generation. In *ACL*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021b. Pretrained language models for text generation: A survey. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021*.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *AAAI 2019, IAAI 2019, AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6786–6793. AAAI Press.

Robert Logan, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2267–2277. Association for Computational Linguistics.

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.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics.

Maja Popovic. 2015. chrf: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 392–395. The Association for Computer Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020a. Modeling global and local node contexts for text generation from knowledge graphs. *Trans. Assoc. Comput. Linguistics*, 8:589–604.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schutze, and Iryna Gurevych. 2020b. Investigating pre-trained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7987–7998. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. 2020. Improving text-to-text pre-trained models for the graph-to-text task. In *Proceedings of the 3rd WebNLG Workshop on Natural Language Generation from the Semantic Web (WebNLG+ 2020), Dublin, Ireland (Virtual). Association for Computational Linguistics*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.