# Syntax Controlled Knowledge Graph-to-Text Generation with Order and Semantic Consistency

**Jin Liu[1], Chongfeng Fan[1], Fengyu Zhou[1], Huijuan Xu[2]**

[1] Shandong University
[2] Pennsylvania State University

{202120638,202114753,zhoufengyu} @mail.sdu.edu.cn,
hkx5063@psu.edu

## Abstract

The knowledge graph (KG) stores a large amount of structural knowledge, while it is not easy for direct human understanding. Knowledge graph-to-text (KG-to-text) generation aims to generate easy-to-understand sentences from the KG, and at the same time, maintains semantic consistency between generated sentences and the KG. Existing KG-to-text generation methods phrase this task as a sequence-to-sequence generation task with linearized KG as input and consider the consistency issue of the generated texts and KG through a simple selection between decoded sentence word and KG node word at each time step. However, the linearized KG order is commonly obtained through a heuristic search without data-driven optimization. In this paper, we optimize the knowledge description order prediction under the order supervision extracted from the caption and further enhance the consistency of the generated sentences and KG through syntactic and semantic regularization. We incorporate the Part-of-Speech (POS) syntactic tags to constrain the positions to copy words from the KG and employ a semantic context scoring function to evaluate the semantic fitness for each word in its local context when decoding each word in the generated sentence. Extensive experiments are conducted on two datasets, WebNLG and DART, and achieve state-of-the-art performances. Our code is now public available[1].

## 1 Introduction

Knowledge graphs (KGs) record the common sense knowledge in a structural way and have many potential applications, e.g., question answering (Saxena et al., 2020), recommendation system (Wang et al., 2021) and storytelling (Xu et al., 2021). One typical KG is shown in Figure 1, with circle nodes indicating entities, and the directional edges connecting the head node to the tail node and representing the relation among connected entities. This
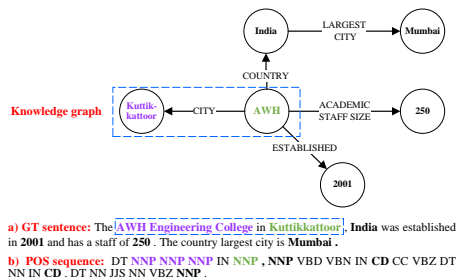


Figure 1: One example KG from DART dataset (Nan et al., 2020). AWH is short for AWH Engineering College. Text a) is the ground-truth sentence, text b) is the POS sequence obtained by NLTK, and most words copied from the KG are nouns. The linearized order of KG should correlate with the information sequence expressed in the corresponding sentence (words indicated in bold in GT sentence), and each decoded word should fit in its local semantic context (e.g. the blue dashed lines).

structural representation in KG is easy for information storage while not convenient for human understanding. In this paper, we focus on the task of KG-to-text generation, which aims to describe an input KG with fluent language sentences in an easy-to-understand way. Compared to the traditional text generation task, KG-to-text generation poses the extra challenge of maintaining the word authenticity in the generated sentence given the input KG. With the word authenticity and sentence fluentness in mind, existing KG-to-text generation methods (Ribeiro et al., 2020a; Koncel-Kedziorski et al., 2019; Gardent et al., 2017) phrase this task as a sequence-to-sequence generation task, where the KG is linearized as input sequence and decoded into sentences, and the word from the KG are selected to be inserted into the decoded sentence with predicted confidence at each time step. However, when the KG is linearized into a sequence, simple heuristic search-based algorithms are usually utilized, e.g., breadth-first search (BFS) or using other pre-defined rules for sorting (Li et al., 2021;

---

[1] https://github.com/LemonQC/KG2Text

1278

Ribeiro et al., 2020a), without considering the word sequence information in the ground-truth sentences. The order inference of the KG is not tightly correlated to the word sequence information in the ground-truth sentences and is generated in a disjoint prestage. The decoded sentence is further conditioned on the linearized KG order which might incur cascaded errors (Cornia et al., 2019). To tackle this problem, we extract the order information from the ground-truth sentence and use this order information to directly supervise the KG order prediction with graph structural local context. In this case, our order prediction component for KG encodes the sentence sequence prior information and will benefit the sentence generation in the follow-up stage.

Also, most existing methods (Li et al., 2021; Koncel-Kedziorski et al., 2019) maintain the word authenticity by maximizing the copy probability of the tokens from the KG, while ignoring the syntactic correctness and semantic relevance. A simple observation from the example in Figure 1, you can find that the POS tags of most words copied from the KG are nouns. Motivated by this observation, we introduce a POS generator to guide the sentence generation process by applying the POS information as additional supervision at each time step and limit the position scope of the word selection from KG. Moreover, to further enhance semantic relevance of generated sentence, we consider the structural information and local semantic information of the KG by designing a semantic context scoring function with sliding windows of different sizes, and combine the semantic context score into the word selection process at each time step of the sentence generation.

In summary, we propose a Syntax controlled KG-to-text generation model with Order and Semantic Consistency, called S-OSC. The main contributions are summarized as follows:

- We propose a learning-based sorting network to obtain the optimal KG description order with graph structural context for more fluent caption generation.

- We enhance the authenticity of generated sentences to the KG through syntactic and semantic regularization. POS tag information is incorporated into the sentence modeling and helps to determine the word selection from KG, together with one additional semantic context scoring function.

- Extensive results on two benchmark datasets indicate that our proposed S-OSC model outperforms previous models and achieves new state-of-the-art performance.

# 2 Related Work

## 2.1 KG-To-Text Generation

KG-to-text generation task has been a hot research topic since the first dataset WebNLG was proposed (Gardent et al., 2017). Recent works for solving this task have two main categories. One category is using graph neural networks (Marcheggiani and Perez-Beltrachini, 2018; Ribeiro et al., 2020b; Li et al., 2021; Cheng et al., 2020) or graph transformers (Koncel-Kedziorski et al., 2019) to directly capture the graph structure information and decode into sentences. E.g., the recent work (Ribeiro et al., 2020b) forms four different encoder architectures for combining local and global node contexts. ENT-DESC (Cheng et al., 2020) introduces multi-graph structure to better aggregate knowledge information. The other category is first linearizing the KG (Ribeiro et al., 2020a; Yang et al., 2020; Gardent et al., 2017; Hoyle et al., 2020) and then formulating a sequence-to-sequence generation task with lineared KG nodes as input to generate sentences. E.g., Distiawan et al. (Distiawan et al., 2018) utilize a fixed tree traversal order to directly flatten the KG into a linearized representation. The structural information of the KG is not preserved when generating the KG order. In this paper, we follow the second general pipeline with local structure encoded in the order generation process.

## 2.2 Knowledge Graph Order Generation

A line of KG-to-text models (Ribeiro et al., 2020a; Braude et al., 2021) tries to generate sentences conditioned on the KG order, where the order generation is especially important as different KG description orders may result in various generated texts. Most previous works focus on graph traversal-based approaches (Flanigan et al., 2016; Gardent et al., 2017; Ke et al., 2021; Li et al., 2021) for KG order generation. (Li et al., 2021) proposes to use a relation-biased breadth first search (RBFS) strategy to linearize the KG. These previous graph traversal-based approaches are heuristic without considering the word sequence information in the ground-truth sentences. Inspired by previous work (Cornia et al., 2019) which generates image caption with optimal object description order, we extract the sequence

information from the ground-truth sentences as supervision and train one order prediction module to generate optimal order. Moreover, our order prediction considers the local graph structure in triplet.

## 2.3 Captioning with POS Tags

POS tags have been used in various text generation (e.g. image captioning and video captioning) to impose the syntactic constraint. In the neural text generation work (Yang and Wan, 2021), the authors propose to use POS guided softmax function as the linguistic prior information for modeling the posterior probabilities of next-POS and next-token, in order to increase text generation diversity. In the image caption, Bugliarello et al. (Bugliarello and Elliott, 2021) claim that incorporating POS tag information in the sentence generation process consistently improves the quality of the generated text. In video captioning, Hou et al. (Hou et al., 2019) propose to define the templates of POS tag sequences to represent the syntactic structure of the generated text. In KG-to-text generation task, we not only use the POS tags to ensure the syntactic correctness of the generated text, but also use the POS tags to constrain the positions to copy words from KGs.

## 2.4 Pre-Trained Language Models

Pre-trained language models (PLMs) on massive corpora, such as BERT (Devlin et al., 2018), BART (Lewis et al., 2019) and T5 (Raffel et al., 2019), have achieved superior performance in various natural language generation tasks, including KG-to-text generation task (Ribeiro et al., 2020a; Peters et al., 2019; Ke et al., 2021). Ribeiro et al. (Ribeiro et al., 2020a) leverage the generation ability of PLMs and use the linearized KG as input to generate texts. We also use the PLM in our method to guarantee the model generalization ability, and at the same time, design extra order prediction and context scoring components to maintain the semantic consistency.

# 3 Approaches

In this section, we first formulate the KG-to-text generation problem setting and then elaborate the proposed S-OSC in detail.

## 3.1 Problem Formulation

Given the input KG $\mathcal{G}$, which is composed of $\{(h_1, r_1, t_1), \cdots, (h_n, r_n, t_n)|h_*, t_* \in \mathcal{E}, r_* \in$
$\mathcal{R}\}$, where $\mathcal{E}$ denotes the entity set and $\mathcal{R}$ represents the relation set, the KG-to-text generation task aims to generate a fluent and reasonable text sequence $\mathcal{T} = < t_1, t_2, \cdots, t_k > (t_k \in \mathcal{V})$, where $\mathcal{V}$ denotes the vocabulary. In this paper, we follow the general pipeline (Ribeiro et al., 2020a; Ke et al., 2021; Ribeiro et al., 2021) of linearizing the input KG into sequence $\mathcal{G}_{linear} = < g_1, g_2, \cdots, g_m >$ consisting of $m$ tokens, and then decoding the KG token sequence into sentences.

## 3.2 Our Proposed S-OSC Model

We propose S-OSC, illustrated in Figure 2, which consists of two main components: one learning-based sorting network for KG, and one copy or prediction selection module for decoding each word in the sentences. The sorting network generates the optimal description sequence for the input KG. Based on the KG order sequence, the sentence decoder generates each word with a certain probability predicted by the copy or prediction selection module to replace the decoded word with the word in the KG. Thus, the model can maintain the word authenticity in the generated sentence compared to the KG. A key innovation in our learning-based sorting network is to utilize the sequence information extracted from the ground-truth sentence to directly supervise the optimal sequence prediction instead of heuristic search in the KG without considering the description sequence prior. Our copy or prediction selection module for decoding each word in the sentences incorporates additional POS syntactic constraint and semantic context consistency scoring function evaluating the semantic fitness of each word in its sliding windows with various sizes. The details of each module in our S-OSC model are illustrated in the following sections.

### 3.2.1 Sorting Network

The description order of the KG will affect the content of the generated sentence. In a worse scenario, poor order may result in the loss of important information (see the example in Figure 7). To overcome the drawback of disjoint learning for KG order generation and sentence generation in the previous heuristic-based methods (Li et al., 2021; Ribeiro et al., 2020a; Yang et al., 2020), we propose a learning-based sorting network with the order supervision extracted from the ground-truth sentence. Notably, our sorting network is based on the features from the structural Triplet Encoder, where the
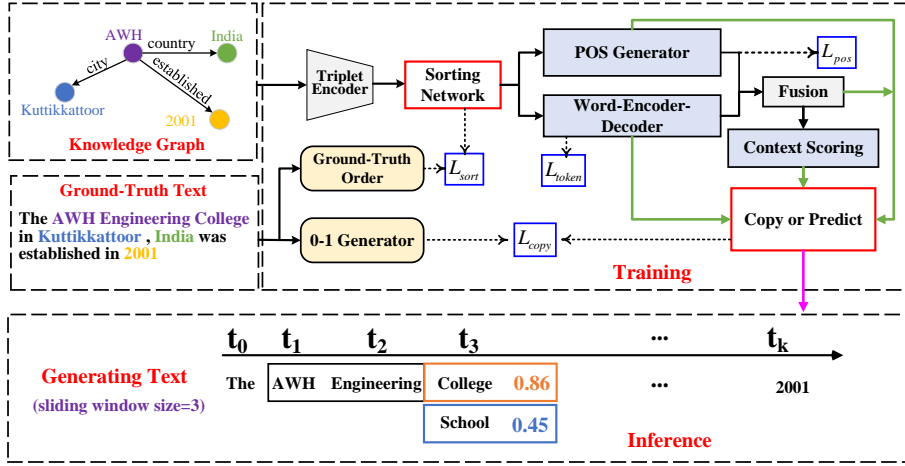
Figure 2: S-OSC model architecture. The pre-trained language model takes the KG as input and extracts head-relation-tail triplet structure features through Triplet Encoder. Then, these triplet features are fed into the Sorting Network to generate one optimal description order under the sequence supervision extracted from ground-truth sentence. Conditioned on the KG order, we further decode the sequence into sentences through Word Decoder and apply additional syntactic supervision through POS Generator. To maintain the word authenticity in the generated sentences, we further design one Copy or Prediction component incorporating the POS syntactic information and semantic context scoring with sliding windows to help determine when to copy tokens from KG, in addition to the word decoding probability from the Word-Encoder-Decoder and the binary classification probability for copy or prediction at each time step.

head-relation-tail triplet structure features are extracted through pre-trained KG embedding method TransH (Wang et al., 2014) and pre-trained language model BART (Lewis et al., 2019). Due to the variable length of the KG, we introduce a placeholder to pad it into fixed length $N$, which also denotes the number of possible position classes. The head-relation-tail triplet structure features $F_{stru}$ are concatenated with the padding $F_{pad}$ and fed through the Fully Connected (FC) layers with softmax classifier $FC_s$ to obtain $S_{matrix}$ and predict the sorting order $S_{order}$.

$$S_{matrix} = FC_s([F_{stru}; F_{pad}])$$
$$S_{order} = \text{argmax}_{row}(S_{matrix}) \quad (1)$$

In this paper, we treat the order prediction task as a classification problem, where N denotes the number of classes (the maximum of the triplets in KG). Thus, we measure the cross-entropy loss between the ground-truth order $G_{order}$ and the sorting order $S_{order}$:

$$L_{sort} = -\sum_{n=0}^{N} \log(S_{matrix}^n) \cdot G_{order}^n \quad (2)$$
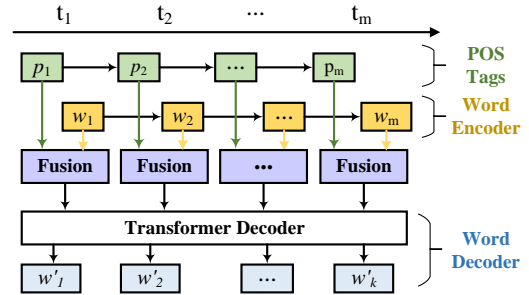


Figure 3: The architecture of the Word-Encoder-Decoder with a POS Generator.

### 3.2.2 Copy or Prediction Network

To maintain the authenticity of KG words in generated sentences, the model needs to selectively copy words from KG words instead of using the predicted words from Word Decoder. Besides directly predicting the copy probability from the hidden state (Koncel-Kedziorski et al., 2019; Li et al., 2021; See et al., 2017), the Copy or Prediction Network in our S-OSC model further enhances the syntactic and semantic consistency of selected words in the decoded sentence through the incorporation of POS generator (shown in Figure 3) and semantic context scoring (shown in Figure 4). Next, we will introduce each module in detail.

**POS Syntactic Constraint** Conditioned on the KG order $G_{order}$, we first linearize the KG by
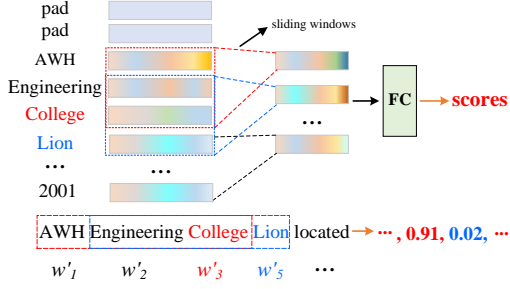
Figure 4: The architecture of semantic context scoring module with sliding windows.

adding the tokens $< Head >, < Relation >, < Tail >$ to the corresponding position for each triplet and obtain the $\mathcal{G}_{linear}$. Then, the Word Encoder and the POS Generator take $G_{linear}$ as their inputs and output the word encoding WI=$\{w_i, i \in 1 \cdots m\}$ and POS tag encoding PI=$\{p_i, i \in 1 \cdots m\}$, respectively. Then, the token encoding $w_i$ and POS tag encoding $p_i$ are combined in the fusion module to get the updated token encoding $w_i$.

$$w_i = \text{LN}(FC([w_i; p_i]) + w_i), \qquad (3)$$

where LN denotes the layer normalization. The updated token encoding $w_i$ after fusing is decoded into sentence WI$' = \{w'_i, i \in 1 \cdots k\}$ in Word Decoder.

POS generator is supervised through POS tags pre-extracted from the sentence. The loss function is formulated as:

$$L_{pos} = -\sum_{l=1}^{M} \log(P_{gen}(p_l|p_1, \cdots, p_{l-1}; G_{order})), \qquad (4)$$

where $P_{gen}$ denotes the predicted probability from POS generator. Similarly, the objective of Word-Encoder-Decoder is as follows:

$$L_{token} = -\sum_{j=1}^{K} \log(W_{gen}(w'_j|w'_1, \cdots, w'_{j-1}; \text{WI}')), \qquad (5)$$

where $W_{gen}$ denotes the predicted probability of each word token.

**Semantic Context Scoring** Besides the syntactic constraint for copied words, we also design one semantic context scoring component, illustrated in Figure 4, to evaluate the semantic consistency of copied or predicted words in the sliding windows. Sliding windows are generated for each word to provide the local context, e.g., the sliding window

size is set to 3 in Figure 4. Besides, padding is needed for the first several words when forming the sliding windows. Word features in the sliding window are contacted to get the context information $F_{context}$, and are fed into the FC layers to obtain the semantic score $X_{semantic}$.

$$X_{semantic} = \sigma(\text{FC}(F_{context})), \qquad (6)$$

where $\sigma$ denotes the sigmoid function.

**Word Copy Probability Prediction** With our newly introduced POS token embeddings $v_{p_k}$ and the semantic context score $X_{semantic}$, the probability $p_{copy}^k$ for copying words from KG is computed in Eq. 7, and is used in testing time for final selection between predicted words from Word Decoder and words in KG at each time step when generating sentences.

$$t_{copy}^k = \sigma(W_1 v_{w_k} + W_2 v_{p_k} + W_3 s_k + b_{copy}),$$
$$p_{copy}^k = \lambda \cdot X_{semantic} + (1 - \lambda) \cdot t_{copy}^k, \qquad (7)$$

where $W_1$, $W_2$, $W_3$ and $b_{copy}$ are learnable parameters. $v_{w_k}$ represents token embedding and $s_k$ represents the last hidden state of Word-Decoder at each time step. $\lambda$ is a trade-off coefficient and is set as 0.3.

The semantic context scoring module is jointly optimized with copy probability prediction and benefits the copy probability prediction. The copy or prediction loss function is defined as:

$$L_{copy} = -\sum_{k=0}^{K}(y^k \cdot \log(p_{copy}^k) + (1 - y^k) \qquad (8)$$
$$\cdot \log(1 - p_{copy}^k)),$$

where $y^k$ is the ground-truth 0-1 label indicating copying or predicting word at $k - th$ time step, which is generated from KG and the ground-truth sentence (see more details in supplementary material).

Finally, the total training loss $L_{total}$ in our S-OSC model is composed of four components: sorting loss $L_{sort}$(Eq. 2), POS generation loss $L_{pos}$(Eq. 4), word generation loss $L_{token}$(Eq. 5) and copy or prediction loss $L_{copy}$(Eq. 8).

$$L_{total} = L_{token} + \lambda_1 L_{pos} + \lambda_2 L_{sort} + \lambda_3 L_{copy}, \qquad (9)$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ are the trade-off coefficients.

# 4 Experiments

In this section, we report the comparison results with state-of-the-art methods and further analyze the performance of each component in our S-OSC model through ablation studies. We also evaluate our model performance through human evaluation and qualitative analysis.

## 4.1 Datasets

In this paper, two benchmarks: WebNLG (Li et al., 2021) and DART (Nan et al., 2020) are utilized to evaluate the performance of our S-OSC model.

**WebNLG**  WebNLG (Li et al., 2021) is a most widely used dataset in KG-to-text generation task. Each graph is extracted from DBPedia and consists of two to seven triplets. The train/val/test splits are 7362/1389/5427.

**DART**  Compared to WebNLG, DART (Nan et al., 2020) is a larger open-domain dataset, where triples are composed of tree-structured ontology. The train/val/test splits are 30348/2759/5097.

## 4.2 Evaluation Metrics

Following the previous works (Ribeiro et al., 2020a,b; Li et al., 2021) on WebNLG dataset, we adopt four automatic language evaluation metrics for WebNLG dataset, i.e., BLEU-4 (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), Chrf++ (Popović, 2015) and ROUGE-L (Lin, 2004). Following previous works (Nan et al., 2020) on DART dataset, in addition to BLEU-4 (Papineni et al., 2002), we use four additional automatic evaluation metrics, i.e., METEOR (Banerjee and Lavie, 2005), MoverScore (Zhao et al., 2019), BERTScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020).

## 4.3 Implementation Details

In the sorting network, the fixed KG order length $N$ is set to 8 and 10 for WebNLG and DART, respectively. The POS generator operates on the POS sequences parsed from ground-truth sentences via NLTK, and trains from BART-Base pretrained model (Lewis et al., 2019). For Word-Encoder-Decoder, we follow the code in JointGT (Ke et al., 2021) and utilize Bart-Base with self-attention (Shaw et al., 2018). The beam search size for generating sentences in inference time is set to 5. We optimize all the parameters under the supervision of the total loss in Eq. 9 using the OpenAI AdamW optimizer. The loss weights $\lambda_1$, $\lambda_2$

| Datasets | WEBNLG | | | |
|---|---|---|---|---|
| Metrics | B-4 | R-L | CIDEr | Chrf++ |
| Li et al.[†] (Li et al., 2021) | 57.10 | 75.20 | 4.20 | 75.00 |
| Li et al. (Li et al., 2021) | 61.88 | 75.74 | **6.03** | 79.10 |
| GraphWriter (Koncel-Kedziorski et al., 2019) | 45.84 | 60.62 | 3.14 | 55.53 |
| CGE-LW (Ribeiro et al., 2020b) | 48.60 | 62.52 | 3.85 | 58.66 |
| T5-Base (Ribeiro et al., 2020a) | 48.86 | 65.57 | 3.99 | 66.08 |
| BART-Base (Ribeiro et al., 2020a) | 49.81 | 63.10 | 3.45 | 67.65 |
| T5-Large (Ribeiro et al., 2020a) | 58.78 | 68.22 | 4.10 | 74.40 |
| BART-Large (Ribeiro et al., 2020a) | 52.49 | 65.61 | 3.50 | 72.00 |
| JointGT[†] (Ke et al., 2021) | 57.00 | 77.10 | 4.73 | 76.90 |
| S-OSC(ours) | **61.90** | **79.30** | 5.30 | **79.70** |
| S-OSC(ours)-GT | 63.10 | 80.00 | 5.40 | 81.00 |

Table 1: Results for different models on WebNLG dataset. B-4 and R-L are short for BLEU-4 and ROUGE-L, respectively. **Bold** and underline fonts represent the best and the second best performing results. "S-OSC-GT" indicate our "S-OSC" model with ground-truth order in inference (The same term is applied in the following). [†] denotes the results of reproduction. Other cited results are from Li et al. (Li et al., 2021).

and $\lambda_3$ in total training loss (Eq.9) are set to 0.7, 0.4, and 0.3, respectively.

## 4.4 Main Results on WebNLG and DART

We compare our S-OSC model with other state-of-the-art methods on WebNLG and DART. Results on WebNLG are shown in Table 1. It can be observed from Table 1 that our S-OSC model outperforms all the previous methods in three evaluation metrics, i.e., B-4, R-L and Chrf++, except for the CIDEr value being in the second place compared to the best performing CIDEr result reported by the model in Li et al. (Li et al., 2021). Note that our S-OSC model contains one learning-based sorting network supervised by the ground-truth order extracted from KG and the ground-truth sentence. We also report our model's results under the ground-truth order during inference time (denoted as "S-OSC-GT") which serves as the upper bound for our sorting performance. From the comparison of our model with predicted order "S-OSC" and our model with ground-truth order "S-OSC-GT", we can see that our model results with predicted order are close to the results with ground-truth order with the result gaps less than 1.2 points in most metrics, which shows the advantage of our learning-based sorting network for generating KG description order.

Results on DART are shown in Table 2. From previous models' results, we can see that pre-training on the large corpus (e.g., "T5-Large" and "Bart-Large") brings significant result improvement compared to "Seq2Seq-Att" and "End-to-End Transformer". Our S-OSC model further achieves

| Datasets | DART | | | | |
|---|---|---|---|---|---|
| Metrics | B-4 | METEOR | MoverScore | BERTScore | BLEURT |
| Seq2Seq-Att (Nan et al., 2020) | 29.66 | 0.27 | 0.31 | 0.90 | -0.13 |
| End-to-End Transformer (Ferreira et al., 2019) | 27.24 | 0.25 | 0.25 | 0.89 | -0.29 |
| T5-Large (Ribeiro et al., 2020a) | 50.66 | 0.40 | 0.54 | 0.95 | 0.44 |
| BART-Large (Ribeiro et al., 2020a) | 48.56 | 0.39 | 0.52 | 0.95 | 0.41 |
| JointGT (Ke et al., 2021) | 54.24 | **0.44** | **0.64** | **0.96** | **0.59** |
| S-OSC(ours) | **62.01** | 0.43 | **0.64** | **0.96** | 0.49 |
| S-OSC(ours)-GT | 64.53 | 0.44 | 0.66 | 0.96 | 0.51 |

Table 2: The results of different models on DART dataset.

new state-of-the-art results in all five metrics compared to these pre-training based models (e.g., "T5-Large") with B-4 score improved by 11.35 points, METEOR score improved by 0.03 points, Moverscore improved by 0.1 points, BERTScore improved by 0.01 points, and BLEURT improved by 0.05 points. JointGT outperforms all the previous baseline models. Compared with JointGT (Ke et al., 2021), our model can also obtain an improvement of 7.77 points in B-4 score. This result also validates the effectiveness of our model in improving the fluentness and authenticity of the generated sentences with the proposed learning-based sorting network and consistency enhancement under the POS syntactic and semantic context constraints. Similar to the results on WebNLG, our model S-OSC with predicted order can achieve the performance close to that with ground-truth order.

## 4.5 Ablation Study

In this section, we conduct extensive experiments on WebNLG to evaluate various factors in the word copy or prediction during sentence generation, e.g., the POS generator and semantic context (SC) scoring. We compare our full S-OSC model with the following variations: without word copy component and just use the predicted word from Word Decoder at each time step (w/o CP), without POS information in copy probability prediction (w/o POS), without semantic context scoring in copy probability prediction (w/o SC), without POS tag information and semantic context scoring in copy probability prediction (w/o POS and SC) and relying on the last hidden states to predict the copy probability as in previous methods (Koncel-Kedziorski et al., 2019; See et al., 2017). From the results in Table 3, we can observe that: (1) By removing the word copy component and directly taking the predicted word from Word Decoder at each time step (w/o CP), the results drop significantly by 3.95 points in

| Methods | B-4 | R-L | CIDEr | Chrf++ |
|---|---|---|---|---|
| S-OSC | 63.10 | 80.0 | 5.40 | 81.0 |
| w/o CP | 59.15 | 77.8 | 4.99 | 78.0 |
| w/o POS and SC | 59.80 | 77.7 | 5.01 | 78.1 |
| w/o POS | 61.00 | 78.5 | 5.10 | 78.6 |
| w/o SC | 61.06 | 78.9 | 5.21 | 79.1 |

Table 3: Ablation analysis for copy or prediction component in our model on WebNLG. S-OSC here is under the ground-truth order.

B-4, 2.2 points in R-L, 0.41 points in CIDEr and 3 points in Chrf++. (2) We then consider incorporating the copy prediction component, but just rely on the last hidden states to predict the copy probability as in previous methods (Koncel-Kedziorski et al., 2019; See et al., 2017) without POS tag information and semantic context scoring in copy probability prediction (w/o POS and SC). The results improve slightly in all the metrics compared with that of "w/o CP". (3) We then evaluate the effectiveness of POS syntactic information and semantic context scores in improving the quality of the generated sentences by removing each of them at a time.

Compared to full S-OSC model, without POS information in copy probability prediction (w/o POS), the results drop by 2.1 points in B-4, 1.5 points in R-L, 0.3 points in CIDEr and 2.4 points in Chrf++. Compared to full S-OSC model, without semantic context scoring in copy probability prediction (w/o SC), the results drop by 2 points in B-4, 1.1 points in R-L, 0.2 points in CIDEr and 1.9 points in Chrf++. Both "w/o POS" and "w/o SC" improve consistently compared to previous copy policy in "w/o POS and SC".

To further reveal more details about our model, we conduct extra ablation studies regarding the following questions. (1) How does triplet structure encoding in the sorting network help the model compared with direct node encoding without triplet structure context? (2) How does our model perform

| Methods | B-4 | R-L | CIDEr | Chrf++ |
|---------|------|------|-------|--------|
| GT | 63.10 | 80.0 | 5.4 | 81.0 |
| **TS** | 61.90 | 79.3 | 5.3 | 79.7 |
| NS | 60.00 | 77.0 | 5.1 | 78.0 |
| RS | 55.20 | 72.0 | 4.7 | 76.0 |

Table 4: The results of sorting order on WebNLG. TS denotes the triple-level sorting order, NS is the node-level sorting order (Li et al., 2021) and RS represents the random sorting order. The same symbols are as below.
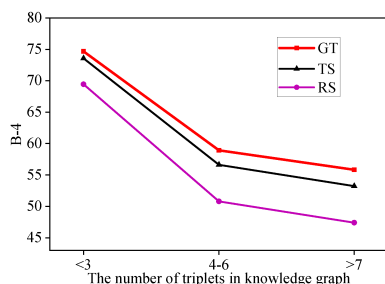


Figure 5: The B-4 results in three groups with different KG sizes on DART dataset for different sorting orders.

for different triplet numbers in KG? (3) How does the sliding window size in semantic context scoring function affect the model performance?

**(1) Triplet structure encoding in sorting network.** To show the effect of triplet structure encoding (triple-level) in our sorting network, we investigate the performance of direct node encoding without triplet structure (node-level), as well as our S-OSC model's results with random sorting order. Results on WebNLG are shown in Table 4 and the upper bound result is shown with ground-truth order (GT). From Table 4, sorting with triple-level sorting order "TS" outperforms random sorting order "RS" significantly, and also outperforms node-level sorting order "NS" by 1.9 points in B-4, 2.3 points in R-L, 0.2 points in CIDEr and 1.7 points in Chrf++, showing the advantage of triplet structure encoding compared to node encoding without triplet structure context. The results on DART dataset show similar trend and are reported in supplementary material.

**(2) Different Knowledge Graph Sizes.** To verify the effect of our S-OSC model performance on different KG sizes, we split the test dataset into three subsets according to the size of the KG, i.e., the number of triples in KG is less than 3, between 4 and 6, and more than 7, and reports results on each subset. DART is selected for experiment due to its wide distribution of KG sizes. The results of B-4 in each subset are plotted in Figure 5, showing that as
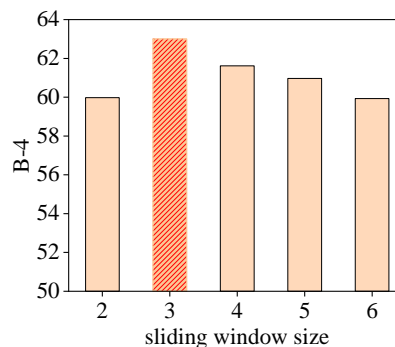


Figure 6: The B-4 results of different sliding windows on WebNLG dataset.

the number of triples in KG increases, the difficulty of sorting and KG-to-text generation increases and the B-4 results generally decrease in all three methods ("GT", "RS" and "TS"). Also, our model's superiority exhibits significant improvement in all three subsets with different KG sizes (comparing "TS" with "RS"). Other metric results show similar trend and are reported in supplementary material.

**(3) Sliding window size in the semantic context scoring.** We plot the B-4 results for different sliding window sizes of semantic context scoring function on WebNLG dataset, shown in Figure 6. From Figure 6, the model achieves the best performance when the sliding window size is 3. Other metric results show similar trend and are reported in supplementary material. Thus, we set the sliding window size to 3 in the experiments.

## 4.6 Human Evaluation

We conduct the human evaluation on WebNLG to further evaluate the generated text. In this paper, we adopt the same human evaluation criteria as Chen et al. (Chen et al., 2019), i.e., Factual correctness including Supp (counting the number of facts that co-exist in the KG and generated text) and Cont (counting the facts in the generated texts missing from or contradicting with KG), Language naturalness including NF (evaluating the accuracy and fluentness of generated sentences). In addition to using absolute score at 5-point metric in NF, we also use relative ranking scores (termed NA). We randomly select 100 knowledge graphs for human evaluation. Five native English speakers volunteer to score all the 100 knowledge graphs. Table 5 reports the results[2]. We can observe that: the generated texts of our S-OSC model are more authentic

---

[2]Cohen's kappa coefficients for the first two factors are 0.79, 0.83,0.75,0.76.

| Datasets | WebNLG | DART |
|---|---|---|
| Knowledge graph | Malaysia (1) → leaderName → Abu Zahar Ujang (2); ethnicGroup → Malaysian Malay (3); ethnicGroup → Malaysian Indian (4) | Antares rocket (1) — MANUFACTURER → Yuzhnoye Design Office (3); FINAL FLIGHT → 2014 10 28 (5); LAUNCH SITE → Mid Atlantic Regional Spaceport Launch Pad 0 (2); MAIDEN FLIGHT → 2013 04 21 (4) |
| Ground-Truth text | Abu zahar ujang is a leader in Malaysia, where there are ethnic groups called Malaysian Malay and Malaysian Indian. | The Antares rocket , made by Yuzhnoye Design Office , had its initial flight April 21 , 2013 and final flight on 10/28/2014 . It was launched at Mid-Atlantic Regional Spaceport Launch Pad 0 . |
| Linearized KG from other methods | Li et al.: 1 → 3 → 2 → 4 | Bart-Large: 1 → 3 → 4 → 5 → 2 |
| Generated text from other methods | Li et al. : The leader of Malaysia is called **jusuf kalla** and Malaysian Malay are an ethnic group there. | Bart-Large: the Yuzhnoye Design Office manufactured the Antares rocket which was launched on april 21 , 2013 at the Mid - Atlantic Regional Spaceport Launch Pad. |
| Linearized KG from ours | 1 → 2 → 3 → 4 | 1 → 3 → 4 → 2 → 5 |
| Generated text from ours | Abu Zahar Ujang is the leader of Malaysia where Malaysian Malay and Malaysian Indian are ethnic groups . | Yuzhnoye Design Office created the Antares rocket whose initial flight took place April 21 , 2013 at Mid Atlantic Regional Spaceport Launch Pad 0 and final flight on 10/28/2014 . |

Figure 7: Two examples from WebNLG and DART. The text in color is corresponding to the KG nodes. The generated texts in red color indicate that the generated texts contract with KG.

| Methods | Supp.↑ | Cont.↓ | NF↑ | NA↑ |
|---|---|---|---|---|
| *ground-truth* | 3.82 | 0.10 | 4.75 | 2.93 |
| Li et al.(Li et al., 2021) | 3.48 | 0.33 | 3.7 | 2.10 |
| S-OSC | 3.77 | 0.15 | 4.25 | 2.48 |

Table 5: The results of human evaluation on WebNLG dataset.

and consistent with KG than the method in Li et al. (Li et al., 2021).

### 4.7 Qualitative Analysis

We show two qualitative examples from WebNLG and DART in Figure 7. From the example of WebNLG, we can see that the previous method in Li et al. (Li et al., 2021) generates the wrong description order with node 3 and node 2 exchanged positions. This causes the generated sentence from Li et al. (Li et al., 2021) containing the wrong text "jusuf kalla" noted in red color and missing the text "Malaysian Indian", while our S-OSC method generates the right order and consistent sentences. In the example from DART, though our S-OSC model generates relative inferior order with node 2 and node 5 exchanged positions, relying on our strong sentence generator with syntactic constraint and semantic consistency constraint, our model is able to generate semantic consistent sentence to the KG with the right words copied from KG. However, in this example, Bart-Large (Ribeiro et al., 2020a) still misses some key word description from KG,

e.g., "the final fight on 10/28/2014" and "Launch Pad 0", though conditioned on the right predicted order, showing the inferior performance of their copy or prediction module in sentence generation.

## 5 Conclusion

This paper proposes a learning-based sorting network to obtain the optimal description order for KG-to-text generation. Additionally, our model incorporates POS generator and semantic context scoring to selectively copy words from KG and improve the word authenticity in generated sentences. Extensive experiments show that our model outperforms previous state-of-the-art approaches. In the future, we will introduce casual inference into the model to further improve the reasoning ability.

## Acknowledgements

# References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Tom Braude, Idan Schwartz, Alexander Schwing, and Ariel Shamir. 2021. Towards coherent visual storytelling with ordered image attention. *arXiv preprint arXiv:2108.02180*.

Emanuele Bugliarello and Desmond Elliott. 2021. The role of syntactic planning in compositional image captioning. *arXiv preprint arXiv:2101.11911*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2019. Few-shot nlg with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

Liying Cheng, Dekun Wu, Lidong Bing, Yan Zhang, Zhanming Jie, Wei Lu, and Luo Si. 2020. Ent-desc: Entity description generation by exploring knowledge graph. *arXiv preprint arXiv:2004.14813*.

Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, control and tell: A framework for generating controllable and grounded captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8307–8316.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bayu Distiawan, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. Gtr-lstm: A triple encoder for sentence generation from rdf data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637.

Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime G Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739.

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*, pages 124–133.

Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. 2019. Joint syntax representation learning and visual cue translation for video captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8918–8927.

Alexander Hoyle, Ana Marasović, and Noah Smith. 2020. Promoting graph awareness in linearized graph-to-text generation. *arXiv preprint arXiv:2012.15793*.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. *arXiv preprint arXiv:2106.10502*.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623*.

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

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. *arXiv preprint arXiv:1810.09995*.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

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. *arXiv preprint arXiv:1909.04164*.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020a. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Leonardo FR Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020b. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

Leonardo FR Ribeiro, Yue Zhang, and Iryna Gurevych. 2021. Structural adapters in pretrained language models for amr-to-text generation. *arXiv preprint arXiv:2103.09120*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*, pages 878–887.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Chunpu Xu, Min Yang, Chengming Li, Ying Shen, Xiang Ao, and Ruifeng Xu. 2021. Imagine, reason and write: Visual storytelling with graph knowledge and relational reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3022–3029.

Zhixian Yang and Xiaojun Wan. 2021. Neural text generation with part-of-speech guided softmax. *arXiv preprint arXiv:2105.03641*.

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 International Workshop on Natural Language Generation from the Semantic Web (WebNLG+).*, pages 107–116.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

# Supplementary Material for
# Syntax Controlled Knowledge Graph-to-Text Generation with Order and Semantic Consistency

**Jin Liu[1], Chongfeng Fan[1], Fengyu Zhou[1], Huijuan Xu[2]**
[1] Shandong University
[2] Pennsylvania State University
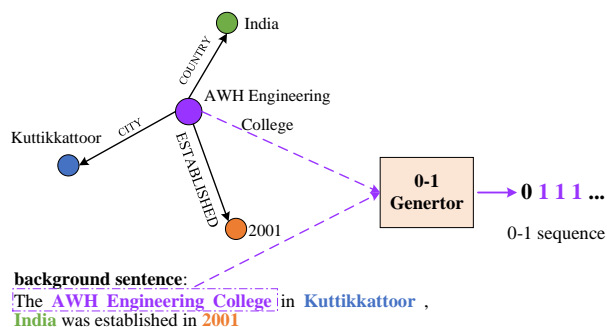{202120638,202114753,zhoufengyu} @mail.sdu.edu.cn,
hkx5063@psu.edu

Figure 1: The steps of obtaining the 0-1 sequence. When the KG node exists in the text, all mentions of the node are masked as 1, otherwise 0.

| Datasets | WEBNLG | | | |
|---|---|---|---|---|
| Metrics | B-4 | R-L | CIDEr | Chrf++ |
| Baselines[†] | 57.00 | 75.20 | 4.20 | 75.00 |
| Li et al.(Li et al., 2021) | 61.88 | 75.74 | **6.03** | 79.10 |
| S-OSC(ours) | **61.90** | **79.30** | 5.30 | **79.70** |

Table 1: Results for different models on WebNLG dataset. B-4 and R-L are short for BLEU-4 and ROUGE-L, respectively. Baseline is the reproduced results of Li et al. (Li et al., 2021). **Bold** and underline represent the best and the second best performing results. (the same term is used below)

## 1 0-1 Generator

In order to make the generated text more consistent with the KG, we use a 0-1 generator to obtain the ground-truth of copy or prediction order, where 0 represents a token from generation and 1 from KG. The steps are as shown in Figure 1.

## 2 Ground-Truth Sorting Order

The description sorting order is essential for the generation process. In this paper, we sort the order according to the triple-level. Taking Figure 1 as an example, we can obtain three triplets, i.e., (AWH Engineering College, COUNTRY, India), (AWH Engineering College, ESTABLISED, 2001) and (AWH Engineering College, CITY, Kuttikkattoor). In the reference sentence, the triple (AWH Engineering College, CITY, Kuttikkattoor) appears first, (AWH Engineering College, COUNTRY, India) appears second and (AWH Engineering College, ESTABLISED, 2001) last. Thus, we can form the order 2,0,1 for the providing KG. Note that, we only record the position of the triplet where it first appeared regardless of many times.

## 3 Code Re-implementation

We re-implement the author's code in Li et al. (Li et al., 2021) as our baseline. It can be observed from Table 1 that our S-OSC model outperforms the baseline and the results of Li et al. reported from their paper in three evaluation metrics,B-4, R-L and Chrf++ metrics. Although Li et al. achieves the best CIDEr score in their reported results, this score is not replicable and our S-OSC still obtains 1.1 point higher than that of the replicated baseline.

## 4 Triplet Structure Encoding in Sorting Network on DART (Nan et al., 2020)

From Table 2, encoding with triple-level sorting order "TS" outperforms random sorting order "RS" significantly, and also outperforms node-level sorting order "NS" by 1.35 points in B-4, 0.01 points in METEOR, 0.02 points in BERTScore and 0.13 points in BLEURT, showing the advantage of triplet structure encoding compared to node encoding without triplet structure context.

## 5 Different Knowledge Graph Sizes

Table 4 shows that when the number of triples in KG increases, the difficulty of sorting and KG-to-text generation increases (see "GT" and "TS") and our model's superiority exhibits more relative im-

| Methods | B-4 | METEOR | MoverScore | BERTScore | BLEURT |
|---------|-----|--------|------------|-----------|--------|
| GT | 64.53 | 0.44 | 0.66 | 0.96 | 0.51 |
| TS | 62.01 | 0.43 | 0.64 | 0.96 | 0.51 |
| NS | 60.66 | 0.42 | 0.64 | 0.94 | 0.38 |
| RS | 58.67 | 0.42 | 0.62 | 0.93 | 0.34 |

Table 2: The results of sorting order on Dart

| Methods | B-4 | R-L | CIDEr | Chrf++ |
|---------|-----|-----|-------|--------|
| With local POS | 63.10 | 80.00 | 5.40 | 81.0 |
| With global POS | 61.28 | 79.32 | 5.28 | 79.2 |

Table 3: Performance comparisons of our model with different POS information (local POS and global POS) on WebNLG dataset.
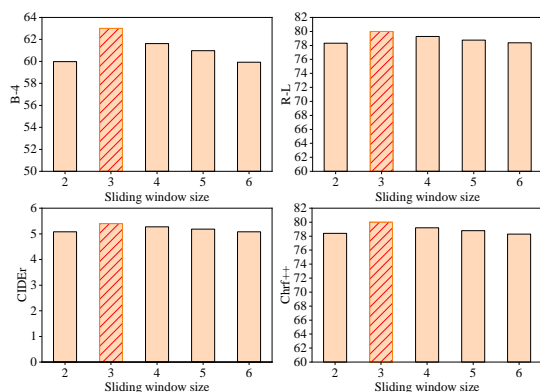


Figure 2: The results of different sliding windows on WebNLG dataset under B-4, R-L, CIDEr and CHrf++ metrics.

provement (comparing "TS" with "NS"). When the size of knowledge graphs is lower than three, these orders have similar performances. This is mainly due to the fact that less triplets have less effect for the model. However, "TS" significantly outperforms "NS" and "RS". Moreover, as the results reported in Table 4, "TS" consistently outperforms "RS" and "NS" in terms of METEOR and MoverScore metrics, and obtains similar performance to "GT" in BERTScore and BLEURT metrics.

## 6 Sliding Window Size in Semantic Context Scoring

As the B-4 result reported in the main paper, the other metric results in Figure 2 also show that the model achieves the best performance when the sliding window size is 3.

## 7 Global and Local POS Information

We further introduce two forms of POS tag information, i.e., the global tag information (the final last hidden state of POS generator) and local tag information (the last hidden states of POS generator at each time step). The results of the comparison are reported in Table 3.

We can observe that the model equipping with local POS information performs better than that of equipping with global information. This is mainly because the local POS information can provide more fine-grained syntactic information. Thus, it can further ensure the authenticity of the generated sentences.

## References

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623*.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

| Datasets | | <3 | | | | 4-6 | | | | >7 | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metrics | METEOR | MoverScore | BERTScore | BLEURT | METEOR | MoverScore | BERTScore | BLEURT | METEOR | MoverScore | BERTScore | BLEURT |
| GT | 0.51 | 0.75 | 0.97 | 0.61 | 0.41 | 0.57 | 0.94 | 0.40 | 0.40 | 0.53 | 0.94 | 0.39 |
| TS | 0.50 | 0.74 | 0.96 | 0.60 | 0.40 | 0.56 | 0.94 | 0.40 | 0.39 | 0.52 | 0.94 | 0.38 |
| NS | 0.49 | 0.72 | 0.97 | 0.58 | 0.39 | 0.54 | 0.94 | 0.37 | 0.38 | 0.49 | 0.94 | 0.38 |
| RS | 0.47 | 0.71 | 0.97 | 0.57 | 0.38 | 0.52 | 0.93 | 0.36 | 0.37 | 0.47 | 0.93 | 0.35 |

Table 4: The results (METEOR, MoverScore, BERTScore and BLEURT metrics) of different sorting orders in the three subsets with difference KG sizes on Dart dataset.

1291