Infobox-to-text Generation with Tree-like PLanning based Attention Network

Yang Bai[1,2*], **Ziran Li**[1,2*], **Ning Ding**[1,2], **Ying Shen**[3] and **Hai-Tao Zheng**[1,2†]

[1]Department of Computer Science and Technology, Tsinghua University
[2]Tsinghua ShenZhen International Graduate School, Tsinghua University
[3]School of Intelligent Systems Engineering, Sun Yat-Sen University
{bai-y18, lizr18}@mails.tsinghua.edu.cn

Abstract

We study the problem of infobox-to-text generation that aims to generate a textual description from a key-value table. Representing the input infobox as a sequence, previous neural methods using end-to-end models without order planning suffer from the problems of incoherence and inadaptability to disordered input. Although recent planning-based models can make some effects, these methods depend on static order-plan to guide generation, which may cause error propagation between planning and generation. To address these issues, we propose a Tree-like PLanning based Attention Network (Tree-PLAN) that leverages both order planning and dynamic tuning to facilitate infobox-to-text generation. We first apply a pointer network to obtain a preliminary order-plan of the input. A novel tree-like tuning encoder is then designed to dynamically tune the order-plan by merging the most relevant attributes together layer by layer. Sets of experiments conducted on two datasets show that our model not only outperforms previous methods on both automatic and human evaluation, but also has better adaptability to disordered input.

1 Introduction

Generating textual descriptions from structured data is a significant and challenging task, which can help people understand the key information from the complex non-linguistic data better. In this paper, we focus on generating fluent, faithful and logically coherent description from an infobox with a set of attributes, each of which can be regarded as a key-value pair. Figure 1 shows an example of writing a description from a given infobox which is in the form of a key-value table.

Previous works using neural models based on encoder-decoder architecture treat the task as an end-to-end learning problem [Lebret *et al.*, 2016; Mei *et al.*, 2016; Wiseman *et al.*, 2017; Liu *et al.*, 2019b]. Some works also exploit the structure information for better input representation [Jain *et al.*, 2018; Liu *et al.*, 2018; Gong *et al.*, 2019], or model content selection for generation [Konstas and Lapata, 2012; Kim and Mooney, 2010; Mei *et al.*, 2016; Shao *et al.*, 2019]. Since all these neural methods represent the input data as a sequence, the order of the input data has a great effect on guiding the generation of descriptions. However, these methods pay little attention to the order-planning and thus have difficulty capturing the order information of the input. In our preliminary experiments, we find that models without explicit order-planning have an unstable performance on disordered input.

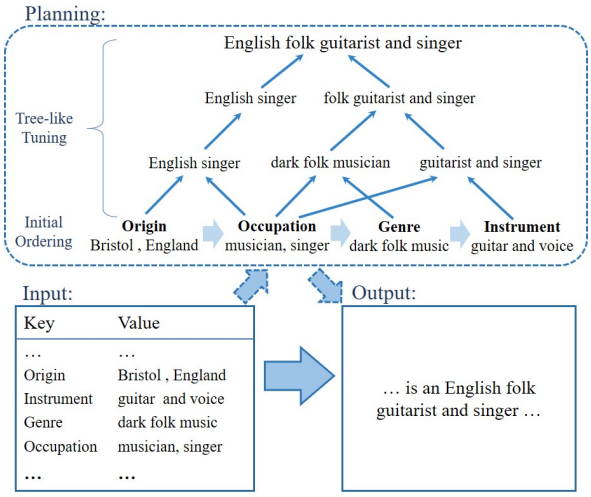

Figure 1: An example of Infobox-to-text generation with tree-like planning.

In a real-world scenario, disordered input data are common and more in line with the practical application. Therefore, explicitly modeling the order-planning to guide the generation is of great necessity. There have been some methods taking the order-planning into account. Specifically, Sha *et al.* [2018] use a link matrix to determine the local order between different attributes while Puduppully *et al.* [2019] employ a pointer network to order the records of the input table. Based on Puduppully's work, Trisedya *et al.* [2020] further design a plan-based bag of tokens attention to handle disordered input. However, these planning methods still face the following problems. (1) Due to the complexity of the input data, the planning results cannot be guaranteed to be perfect, which

*indicates equal contribution.
†Corresponding author: zheng.haitao@sz.tsinghua.edu.cn.

may cause error propagation between planning and generation such as wrong order and missing information. (2) Worse still, the planning stage only makes a static order-plan which is unchangeable once determined. Consequently, the problems of error propagation may get severer because their generation is directly guided by the static order-plan.

Intuitively, if we take the obtained order-plan as an initial plan and utilize the dependency relation among different attributes to dynamically tune it, we can obtain a more reliable and coherent plan to guide the generation. An example is shown in Figure 1 which aims to generate a description of a person from a set of attributes {*i.e. Origin, Instrument, Genre, Occupation, etc.*}. We first order the attributes to an approximately proper order as the initial order-plan. Then considering the relation between different attributes, we reorganize them and merge different attributes layer by layer like a tree, to tune the initial order-plan. At each layer, the most relevant attributes are merged together. Finally, a well-organized plan is obtained and can be used to guide the generation of a more logically coherent description (*... an English folk guitarist and singer*).

To this end, we propose a **Tree**-like **PL**anning based **A**ttention **N**etwork (**Tree-PLAN**) which leverages both order-planning and dynamic tuning to facilitate infobox-to-text generation. After representing the attributes in the input infobox with an attention based encoder, we first apply a pointer network to preliminarily order the input attributes and obtain an initial order-plan. As mentioned above, the initial order-plan is not enough to directly guide the generation, a tree-like tuning encoder is thus utilized to dynamically tune the initial order-plan for better planning. Specifically, we implement a merging attention mechanism to capture the dependency relations among different attributes to merge the most relevant attributes together layer by layer. At each layer, we design a hierarchical tuning mechanism that tunes the initial order-plan on both word-level and attribute-level. Finally a dual attention based decoder is employed to leverage both attribute-level attention and word-level attention to generate textual descriptions guided by the determined plan.

We conduct sets of experiments on two real-world datasets, which aim to generate a textual description of a person (or restaurant) from a given infobox. The experimental results show that our model outperforms state-of-the-art methods on automatic evaluation metrics and has better adaptability to disordered input. We also implement qualitative human evaluation to further estimate the quality of our model. The results indicate that our model can generate fluent, faithful and logically coherent descriptions.

2 Related Work

Traditional methods for data-to-text generation follow a pipeline of modules including *content selection*, *sentence planning* and *surface realization* [Barzilay and Lapata, 2005; Barzilay and Lapata, 2006; Liang *et al.*, 2009]. Most recent works use end-to-end neural networks to generate textual descriptions directly from the input data or focus on exploiting the structure of data for better representation [Mahapatra *et al.*, 2016; Wiseman *et al.*, 2017; Kaffee *et al.*, 2018; Nie *et al.*, 2018; Liu *et al.*, 2018]. Some works also implement hierarchical attention mechanism to model the structure of tables on multiple levels [Jain *et al.*, 2018; Liu *et al.*, 2019a] or different dimensions [Gong *et al.*, 2019]. To solve the problem of information loss, Liu *et al.* [2019b] propose a force attention method to force the generator to focus on more attributes of the input infobox.

Various studies have been conducted to model content planning explicitly to guide generation including rule-based planning [Konstas and Lapata, 2013] or content selection [Konstas and Lapata, 2012; Kim and Mooney, 2010; Mei *et al.*, 2016; Shao *et al.*, 2019]. Despite generating fluent and grammatically correct descriptions on an ordered input, these models have an unstable performance on disordered input without explicit order-planning. To model the order planning, Sha *et al.* [2018] design a link-based attention to capture the order information of input items while Puduppully *et al.* [2019] implement a pointer network to order the input data to guide the generation. Further, Trisedya *et al.* [2020] design a plan-based bag of tokens attention to handle the disordered input. Differing from these approaches, we propose a tree-like planning method to model both order-planning and dynamic tuning for better order representation.

For dynamic tuning, we employ a tree-like attention encoder that integrates tree structure to multi-head attention. Similar strategy is applied to grammar induction with Tree Transformer [Wang *et al.*, 2019]. However, the Tree Transformer only calculates constituent attention between two neighboring words while our model computes a constrained attention over all the input attributes.

3 Methodology

The input of our model is a set of attributes $x = \{a_1, a_2, ...a_n\}$, each of which can be regarded as a key-value pair. The output of our model is the textual description of the input data with a sequence of words $y = \{y_1, y_2, ..., y_T\}$.

As shown in Figure 2, the architecture of Tree-PLAN is composed of the follow three parts: (1) **Representation**, where a set of multi-head attention encoders are utilized to represent the values of attributes in the input infobox, then each attribute is represented as a weighted sum of the associated values. (2) **Planning**, where a pointer network is applied for preliminary order-planning and further a tree-like tuning encoder is designed to dynamically tune the initial order-plan. (3) **Generation**, which implements a dual attention to decode and generate textual descriptions from the determined plans.

3.1 Representation

The input infobox is a set of attributes represented as key-value pairs $a_i = \langle k_i, v_i \rangle$. For each attribute a_i, its value v_i is flatten as a phrase $v_i = \{w_{i,1}, w_{i,2}, ..., w_{i,m}\}$ where $w_{i,j}$ is the j-th word and m is the length of v_i. We further represent each single word as a key-value pair $v_{i,j} = \langle k_{i,j}, w_{i,j} \rangle$ according to the position of the word in the phrase where $k_{i,j} = [k_i; p_{i,j}]$ and $p_{i,j} = j$.

We first embed all the keys, positions and words into vectors (denoted as $\boldsymbol{k^e}$, $\boldsymbol{p^e}$ and $\boldsymbol{w^e}$) then utilize a set of multi-head self-attention [Vaswani *et al.*, 2017] encoders to encode

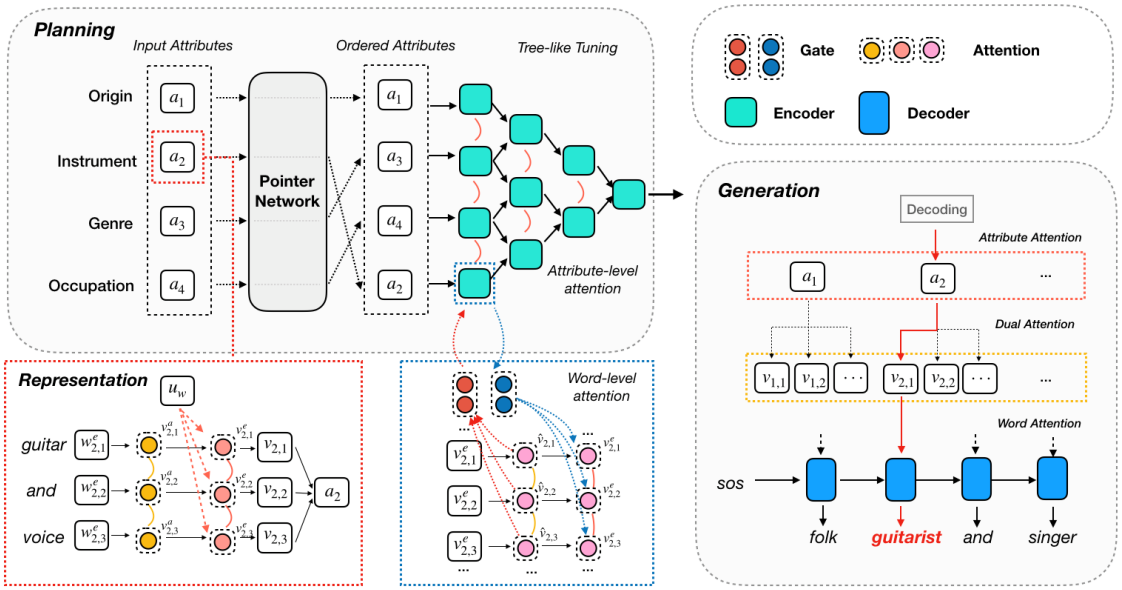

Figure 2: The architecture of Tree-PLAN.

the value of each attribute:

$$\boldsymbol{v}_i^a = \text{Multihead}(\boldsymbol{w}_i^e, \boldsymbol{w}_i^e, \boldsymbol{w}_i^e), \tag{1}$$

where $\boldsymbol{w}_i^e \in \mathbb{R}^{m \times d_w}$ are the words of the i-th attribute and d_w is the dimension of word vectors. Multihead takes queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ as input and calculate as follows:

$$\text{Multihead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = [\boldsymbol{z}^1; ...; \boldsymbol{z}^H]\boldsymbol{W}_o, \tag{2}$$

$$\boldsymbol{z}^h = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_q^h, \boldsymbol{K}\boldsymbol{W}_k^h, \boldsymbol{V}\boldsymbol{W}_v^h), \tag{3}$$

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{4}$$

where $\boldsymbol{W}_o \in \mathbb{R}^{d_w \times d_w}$, $\boldsymbol{W}_q^h, \boldsymbol{W}_k^h, \boldsymbol{W}_v^h \in \mathbb{R}^{d_k \times d_k}$ are parameters, $d_k = d_w/H$, H is the number of attention heads, $[;]$ represents the concatenation operation.

Considering that different word in the same attribute may not contribute the same, an attention mechanism is implemented to estimate the contribution of each word in the same attribute. The contribution weight is calculated as follows:

$$\boldsymbol{u}_{i,j} = \tanh(\boldsymbol{v}_{i,j}^a \boldsymbol{W}_v + \boldsymbol{b}_v), \tag{5}$$

$$\alpha_{i,j} = \frac{\exp(\boldsymbol{u}_{i,j}^\top \boldsymbol{u}_v)}{\sum_j \exp(\boldsymbol{u}_{i,j}^\top \boldsymbol{u}_v)}, \tag{6}$$

where $\boldsymbol{u}_v \in \mathbb{R}^{d_w}$ is the word context vector which is randomly initialized and learned during the training process. $\boldsymbol{W}_v \in \mathbb{R}^{d_w \times d_w}$ and $\boldsymbol{b}_v \in \mathbb{R}^{d_w}$ are parameters, $\alpha_{i,j}$ represents the contribution of $\boldsymbol{v}_{i,j}$. Then we represent the attribute a_i as the sum of $\boldsymbol{v}_i^a$ weighted by α_i:

$$\boldsymbol{a}_i^e = \sum_j \alpha_{i,j} \boldsymbol{v}_{i,j}^a. \tag{7}$$

Finally we get the representation of the words $\{\boldsymbol{v}_i^e\}_{i=1}^n$ and attributes $\boldsymbol{a}^e$.

3.2 Planning

After representing the input infobox, we implement a planning-based encoder to model order planning of the input. The planning process can be divided into two stages: attribute-aware order-planning and tree-like tuning.

Attribute-aware Order-planning

An attribute-aware order-planning is designed to organize the attributes in the input infobox into a reasonable order. A plan $z = \{z_1, z_2, ..., z_{|z|}\}$ is a sequence of the input attributes which are in a new order. Since the output of the order-planning stage corresponds to positions in the input sequence, we apply an attention based pointer network similar to [Wang and Wan, 2019] which is also an encoder-decoder architecture to point to the input attributes.

First a multi-head attention encoder is used to encode the input attributes then a multi-head attention decoder is applied to decode. At each decoding step t, we get the hidden state $\boldsymbol{h}_t$, then the probability $P(z_t|z_{<t}, x)$ is computed as an attention over the input attributes:

$$P(z_t = a_j|z_{<t}, x) = \frac{\exp(\boldsymbol{h}_t^\top \boldsymbol{W}_p \boldsymbol{a}_j^e)}{\sum_i \exp(\boldsymbol{h}_t^\top \boldsymbol{W}_p \boldsymbol{a}_i^e)}, \tag{8}$$

where $\boldsymbol{W}_p \in \mathbb{R}^{d_w \times d_w}$ are parameters

According to the calculated probability distribution, we get the pointer index over the input attributes:

$$\hat{I}_t = \arg\max_j P(z_t = a_j|z_{<t}, x). \tag{9}$$

where $\hat{I}$ represents the index of the ordered attributes. Therefore we can then reorder all attributes (including all keys and values) to a proper order following $\hat{I}$.

We assume that the order in which the attributes appear in the description is the golden order of input attributes. Following this rule, we automatically annotate the order of each

attribute in the input data to get the gold plan and train this stage with supervision, which aims to minimize the negative log-likelihood of the gold plan:

$$\mathcal{L}_1 = -\sum_{(x,z)\in\mathcal{D}}\sum_{t=1}^{|z|}\log P(z_t|z_{<t},x), \tag{10}$$

where $\mathcal{D}$ represents all the training examples including input data, golden plans and target sentences.

The ordering stage ends when a special symbol *eos* is pointed. Noticing that the ordering stage could end before all items are pointed, which may cause information missing, we append the missing attributes in the back of the ordered attributes and obtain an initial order-plan $z = \{\boldsymbol{a}, \langle \boldsymbol{k}, \boldsymbol{p}, \boldsymbol{v}\rangle\}$ including the ordered attributes with the corresponding keys, positions and values.

Tree-like Tuning Encoder

Since the initial order-plans could not be perfect, a dynamic tuning encoder is further designed to dynamically tune the order-plan for better planning, which aims to capture the dependency relations among different attributes and merge the most relevant attributes layer by layer. As shown in Figure 2, at each layer we employ a hierarchical tuning mechanism that tunes the order-plan on both *attribute-level* and *word-level*.

To this end, an extra constraint is added to attention heads and encourages the heads to follow the tree structure. Here we refer to the constraint as *merging attention*. To make each attribute attend to its most relevant one, the attention probability matrix A is modified with the merging attention:

$$A = C * \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right), \tag{11}$$

where $\boldsymbol{Q} = \boldsymbol{W}_q\boldsymbol{a}$, $\boldsymbol{K} = \boldsymbol{W}_k\boldsymbol{k}$, $\boldsymbol{W}_q$ and $\boldsymbol{W}_k \in \mathbb{R}^{d_w\times d_w}$ are parameters, $\boldsymbol{k}$ are the keys of attributes. C is the merging attention and $C_{i,j} = C_{j,i}$ represent the dependency relation between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. Higher value of $C_{i,j}$ means $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are encouraged to merged at this layer.

The merging attention is calculated with attributes $\boldsymbol{a}$ and keys $\boldsymbol{k}$ and is updated layer by layer:

$$\hat{C}^l_{i,j} = \begin{cases} \dfrac{\exp(\boldsymbol{a}_i^\top \boldsymbol{W}_c \boldsymbol{k}_j)}{\sum_j \exp(\boldsymbol{a}_i^\top \boldsymbol{W}_c \boldsymbol{k}_j)} & i \neq j, \\ 0 & i = j, \end{cases} \tag{12}$$

$$\tilde{C}^l_{i,j} = \sqrt{\hat{C}^l_{i,j} * \hat{C}^l_{j,i}}, \tag{13}$$

$$C^l_{i,j} = C^{l-1}_{i,j} + (1 - C^{l-1}_{i,j}) * \tilde{C}^l_{i,j}, \tag{14}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{d_w\times d_w}$ are parameters, l represents the current layer, Eq.(13) is to make sure $C^l_{i,j}$ and $C^l_{j,i}$ are the same.

The tuning encoder is an N-layer tree structure where C^0 is initialized to 0 and Eq.(14) guarantees that C^l is larger than C^{l-1}. Furthermore, considering the special key-value structure of the attributes, each layer is designed as a hierarchical architecture for better planning, where two attention blocks are employed to update the representation of the order-plans on both word-level and attribute-level.

For **attribute-level**, we utilize the computed attention matrix A to encode the represented attribute $\hat{\boldsymbol{a}} = \boldsymbol{A}\boldsymbol{a}$. Then $\hat{\boldsymbol{a}}$ is used to update the representation of each word to $\hat{\boldsymbol{v}}$ via a gate mechanism.

For **word-level**, the updated word $\hat{\boldsymbol{v}}$ is encoded by a multi-head attention layer. Then a content selection mechanism same as Eq.(5-7) is applied to update the representation of the corresponding attribute.

Similar as the transformer encoder, each layer also contains a feedforward sub-layer and a layer normalization. After an N-layer merging and update, the input infobox is encoded on both attribute-level $\hat{\boldsymbol{a}}$ and word-level $\hat{\boldsymbol{v}}$.

3.3 Generation

As the input infobox has been encoded on both attribute-level and word-level, a dual attention based decoder is then applied to decode and generate textual descriptions with the guidance of the determined plans.

The structure of the decoder is similar to the Transformer decoder. At each decoding step t, a multi-head self-attention is first utilized to capture the dependency from the generated words $\boldsymbol{y}_{<t}$ and obtain the hidden state $\boldsymbol{h_t}$. Then, to leverage the information of both attribute-level and word-level of the input data, we employ the dual attention [Gong *et al.*, 2019] to first choose the most relevant attribute based on the key of the attribute then attend to the words in the attribute and obtain the modified attention weights γ. Then we employ another multi-head attention layer to update the hidden states:

$$\hat{\boldsymbol{h}}_t = \mathrm{Multihead}_\gamma(\boldsymbol{h}_t, \boldsymbol{k}^v, \hat{\boldsymbol{v}}), \tag{15}$$

where $\mathrm{Multihead}_\gamma$ is calculated based on the modified attention as mentioned above, $\boldsymbol{k}^v$ are the keys of the words, $\boldsymbol{k}^v_{i,j} = \boldsymbol{k}_i + \boldsymbol{p}_{i,j}$. After a linear layer with a softmax activation, the probability of generated results y_t is obtained as:

$$P(y_t|y_{<t}, z, x) = \mathrm{softmax}(\hat{\boldsymbol{h}}_t \boldsymbol{W}_y), \tag{16}$$

where $\boldsymbol{W}_y \in \mathbb{R}^{d_w\times\mathcal{V}}$ is the word embedding matrix and $\mathcal{V}$ is the vocabulary size.

The goal of the generation stage is to minimize the negative log-likelihood of the sentences in the training set:

$$\mathcal{L}_2 = -\sum_{(x,z,y)\in\mathcal{D}}\sum_{t=1}^{T}\log P(y_t|z,x), \tag{17}$$

where T is the length of the target sentence.

3.4 Training

We train our model end-to-end with a joint learning of both planning and sentence generation by aggregating the losses over the two stages:

$$\mathcal{L} = \lambda\mathcal{L}_1 + (1-\lambda)\mathcal{L}_2, \tag{18}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are calculated by Eq. (10) and Eq. (17), λ is hyper parameter.

4 Experiment

4.1 Experimental Setups

Dataset and Metrics

We conduct experiments on two datasets: (1) WIKIBIO aims to generate the first sentence of a biography from a given Wikipedia infobox. (2) E2E [Novikova *et al.*, 2017] aims to generate descriptions of restaurants from dialogue act-based meaning representations. We follow the previous works [Lebret *et al.*, 2016; Liu *et al.*, 2018; Liu *et al.*, 2019a; Liu *et al.*, 2019b] that use BLEU and ROUGE as the automatic evaluation metrics.

Training Configuration

During experiments we set the dimension of word vectors and hidden state $d_w = 300$ and the number of heads $H = 6$. We choose the most frequent $30,000$ words in the training set as the vocabulary of WIKIBIO. A copy mechanism is applied to replace the unknown words `UNK` with the most likely word in the input data according to the attention distribution. We train the two stages separately for 10 epochs then jointly for 40 epochs. During joint training we set $\lambda = 0.4$.

4.2 Baselines

We compare Tree-PLAN with some strong baseline models as follows, where the former three do not perform planning and the latter three are planning based:

TableNLM [Lebret *et al.*, 2016] is a neural language model which integrates field and position embedding into the data representation.

Struct-aware [Liu *et al.*, 2018] is a seq2seq architecture with a gate mechanism to introduce field information and a dual attention to incorporate the attribute information.

FA+RL [Liu *et al.*, 2019b] is a neural model with force attention and reinforcement learning to force the generator to attend to more attributes of the input infobox.

Order-plan [Sha *et al.*, 2018] is a seq2seq model where a link matrix is designed to model the order of the attributes for infobox-to-text generation.

PHVM [Shao *et al.*, 2019] is a planning-based hierarchical variational model with a high-level planning and a low-level realization for long and diverse text generation.

NCP+BTA [Trisedya *et al.*, 2020] is a neural content-planning based model with bag of tokens attention which uses a joint learning of order-planning and sentence generation.

4.3 Overall Results

Table 1 shows the results of automatic evaluation. Tree-PLAN outperforms all the baselines on both BLEU and ROUGE scores (about $1.62/1.28$ increase on BLEU/ROUGE compared to the state-of-the-art method), indicating that our proposed model can truly facilitate the infobox-to-text generation. The improvements (about $2.42/2.32$ increase on BLEU/ROUGE) compared to the model without planning demonstrate that the proposed order-planning method ia able guide the model to generate higher quality descriptions.

Without attribute-level ordering, our model gets a $1.14/1.24$ decrease on BLEU/ROUGE, which proves that explicit order-planning is able to improve the performance of our model. Moreover, the decline of performance without tree-like dynamic tuning (about 0.66/0.45 decrease on BLUE/ROUGE) illustrates that the tuning mechanism can tune the initial order-plan for better planning, thus improving the performance of our model.

Model	WIKIBIO		E2E	
	BLEU	ROUGE	BLEU	ROUGE
TableNLM	34.70	25.80	–	–
Struct-aware	44.89	41.21	65.77	66.70
FA+RL	45.47	41.54	66.10	67.69
Order-plan	43.91	37.15	–	–
PHVM	44.13	40.37	66.34	68.10
NCP+BTA	45.46	40.31	–	–
Tree-PLAN	**47.09**	**42.82**	**67.45**	**70.08**
− tree	46.43	42.37	67.01	68.75
− order	45.95	41.58	66.79	68.47
− plan	44.67	40.50	66.17	67.72

Table 1: Automatic evaluation results of our models and baselines on WIKIBIO and E2E datasets.

Model	Grammar ↑	Faithful ↑	Coherent ↑
PHVM	3.83	3.36	3.44
Struct-aware	3.92	3.43	3.54
FA+RL	3.88	3.47	3.49
Tree-PLAN	**3.94**	**3.59**	**3.69**
Reference	4.13	3.71	3.86

Table 2: Human evaluation results of different models on WIKIBIO where scores range from 0 to 5.

4.4 Human Evaluation

Since BLEU and ROUGE are calculated based on n-gram matching, automatic evaluation is not enough to evaluate the quality of our model. Therefore we implement human evaluation to evaluate on three aspects: **Grammar** (whether a generated sentence is fluent without grammatical error), **Faithful** (whether the output is faithful to input), and **Coherent** (whether a sentence is logically coherent and the order of expression is in line with human writing habits). We randomly select 300 samples with the descriptions generated by three well-performing baselines and our proposed model. We invite five annotators with sufficient background knowledge to score the given generated descriptions.

The results are reported in Table 2, showing that Tree-PLAN outperforms other models on the three metrics. We find that all the reported models get high scores on Grammar, which illustrates that neural encoder-decoder models are able to generate fluent and grammatically correct descriptions. Compared to other reported models, Tree-PLAN gets a remarkable improvement on Faithful and Coherent (about 0.12 and 0.15 increase), indicating that the proposed planning method can guide to generate more faithful and logically coherent descriptions. The results of human evaluation demonstrate that Tree-PLAN is able to generate more fluent, faithful and logically coherent descriptions.

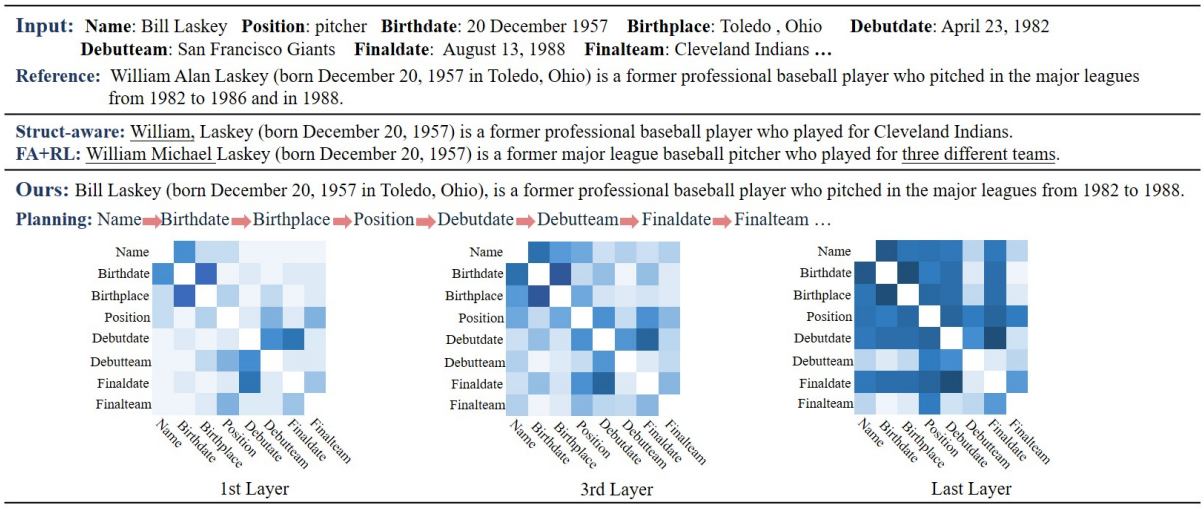

Figure 3: Descriptions generated by different models and the plan of our model (order-plan and the merging attention heat maps).

Model	Max ↑	Min ↑	Mean ↑	Dev ↓
Struct-aware	45.24	43.67	44.19	0.6876
FA+RL	45.87	44.38	44.73	0.5205
NCP	43.75	42.90	43.28	0.3687
PHVM	44.83	43.34	43.64	0.4142
Tree-plan	**47.18**	**46.61**	**46.83**	**0.2213**
– tree	46.57	45.87	46.30	0.3361
– order	46.52	45.34	45.95	0.5009
– plan	45.95	43.45	44.49	0.5947

Table 3: The results of disorder experiments on WIKIBIO (maximum, minimum, mean and deviation of BLEU scores).

4.5 Evaluation of Planning

To further demonstrate the effectiveness of the proposed order-planning, we conduct disorder experiments to evaluate the adaptability to disordered input of Tree-PLAN. We disorder the input attributes and get ten different disordered datasets (including a gold order). We test some baselines and Tree-PLAN on the ten datasets and report the maximum, minimum, mean and deviation of the BLEU score in Table 3. The results show that the order of the input attributes has a significant influence on the performance of neural models.

Tree-PLAN outperforms all other models with higher maximum, higher minimum, higher mean and lower deviation. Higher maximum, minimum and mean indicate the high-performance of Tree-PLAN while lower deviation demonstrate the adaptability of Tree-PLAN to disordered data, which further proves the effectiveness of the proposed planning methods. Furthermore, models with planning have better adaptability to disordered data by having lower deviations, which proves that planning can not only facilitate infobox-to-text generation but also improve the adaptability of neural models to disordered input.

4.6 Case Study

Figure 3 shows the descriptions generated by different models of the same input infobox from the test set of WIKIBIO. Struct-aware misses some important attributes (Birthplace, Position, etc.) while FA+RL gets a better coverage thanks to its force attention mechanism. However, FA+RL still misses some attributes and suffers from the problems of groundless information. Compared to these two models, Tree-PLAN can generate a more faithful and coherent description.

To further show the effect of our tree-like planning, we visualize the results of planning stage in Figure 3 including the initial order-plan and merging attention heat maps. The result of order-plan shows that the multi-head attention based pointer network can generate a approximately proper order-plan. The attention heat maps illustrate how the tuning stage works. The relevant attributes are merged to larger ones layer by layer (e.g. Birthdate and Birthplace are merged at the first layer where the attention weight between the two attributes is high). And finally the input attributes are all merged together in the last layer and a logically coherent plan is obtained.

5 Conclusion

In this paper, we propose a tree-like planning-based attention network (Tree-PLAN) which leverages both static order-planning and dynamic tuning to guide infobox-to-text generation, where a novel tree-like dynamic tuning mechanism is proposed to dynamically tune the static order-plan for better planning. Experiments on two datasets show that Tree-PLAN achieves the state-of-the-art performance and can generate more fluent, faithful and logically coherent descriptions.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 61773229 and 61972219), Shenzhen Giiso Information Technology Co. Ltd., the Basic Research Fund of Shenzhen City (Grand No. JCYJ20190813165003837), Tencent AI Lab Rhino-Bird Focused Research Program (No. JR202032) and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).

References

[Barzilay and Lapata, 2005] Regina Barzilay and Mirella Lapata. Collective content selection for concept-to-text generation. In *HLT/EMNLP*, pages 331–338, 2005.

[Barzilay and Lapata, 2006] Regina Barzilay and Mirella Lapata. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*, page 359–366, New York City, USA, 2006.

[Gong *et al.*, 2019] Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *EMNLP-IJCNLP*, 2019.

[Jain *et al.*, 2018] Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Preksha Nema, Mitesh M. Khapra, and Shreyas Shetty. A mixed hierarchical attention based encoder-decoder approach for standard table summarization. In *NAACL-HLT (2)*, pages 622–627, 2018.

[Kaffee *et al.*, 2018] Lucie-Aimée Kaffee, Hady ElSahar, Pavlos Vougiouklis, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. Learning to generate wikipedia summaries for underserved languages from wikidata. In *NAACL-HLT (2)*, pages 640–645, 2018.

[Kim and Mooney, 2010] Joohyun Kim and Raymond Mooney. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Coling-2010*, pages 543–551, Beijing, China, August 2010.

[Konstas and Lapata, 2012] Ioannis Konstas and Mirella Lapata. Unsupervised concept-to-text generation with hypergraphs. In *HLT-NAACL*, pages 752–761, 2012.

[Konstas and Lapata, 2013] Ioannis Konstas and Mirella Lapata. Inducing document plans for concept-to-text generation. In *EMNLP*, pages 1503–1514, 2013.

[Lebret *et al.*, 2016] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213, 2016.

[Liang *et al.*, 2009] Percy Liang, Michael Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *ACL-IJCNLP*, pages 91–99, Suntec, Singapore, August 2009.

[Liu *et al.*, 2018] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. In *AAAI*, pages 4881–4888, 2018.

[Liu *et al.*, 2019a] Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *AAAI*, pages 6786–6793, 2019.

[Liu *et al.*, 2019b] Tianyu Liu, Fuli Luo, Pengcheng Yang, Wei Wu, Baobao Chang, and Zhifang Sui. Towards comprehensive description generation from factual attribute-value tables. In *ACL*, pages 5985–5996, Florence, Italy, July 2019.

[Mahapatra *et al.*, 2016] Joy Mahapatra, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. Statistical natural language generation from tabular non-textual data. In *INLG*, pages 143–152, 2016.

[Mei *et al.*, 2016] Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *NAACL*, pages 720–730, San Diego, California, June 2016.

[Nie *et al.*, 2018] Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. Operation-guided neural networks for high fidelity data-to-text generation. In *EMNLP*, pages 3879–3889, Brussels, Belgium, October-November 2018.

[Novikova *et al.*, 2017] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany, August 2017.

[Puduppully *et al.*, 2019] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *AAAI*, pages 6908–6915, 2019.

[Sha *et al.*, 2018] Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. Order-planning neural text generation from structured data. In *AAAI*, pages 5414–5421, 2018.

[Shao *et al.*, 2019] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. Long and diverse text generation with planning-based hierarchical variational model. In *EMNLP-IJCNLP*, pages 3255–3266, Hong Kong, China, November 2019.

[Trisedya *et al.*, 2020] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. Sentence generation for entity description with content-plan attention. In *AAAI-2020*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Wang and Wan, 2019] Tianming Wang and Xiaojun Wan. Hierarchical attention networks for sentence ordering. In *AAAI*, pages 7184–7191, 2019.

[Wang *et al.*, 2019] Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. In *EMNLP-IJCNLP*, pages 1061–1070, Hong Kong, China, November 2019.

[Wiseman *et al.*, 2017] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *EMNLP*, pages 2253–2263, Copenhagen, Denmark, September 2017.