Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, Zhifang Sui
Key Laboratory of Computational Linguistics, Ministry of Education,
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
{tianyu0421, luofuli, xql, shumingma, chbb, szf}@pku.edu.cn

Abstract

Generating natural language descriptions for the structured tables which consist of multiple *attribute-value* tuples is a convenient way to help people to understand the tables. Most neural table-to-text models are based on the encoder-decoder framework. However, it is hard for a vanilla encoder to learn the accurate semantic representation of a complex table. The challenges are two-fold: firstly, the table-to-text datasets often contain large number of attributes across different domains, thus it is hard for the encoder to incorporate these heterogeneous resources. Secondly, the single encoder also has difficulties in modeling the complex attribute-value structure of the tables. To this end, we first propose a two-level hierarchical encoder with coarse-to-fine attention to handle the attribute-value structure of the tables. Furthermore, to capture the accurate semantic representations of the tables, we propose 3 joint tasks apart from the prime encoder-decoder learning, namely *auxiliary sequence labeling task*, *text autoencoder* and *multi-labeling classification*, as the auxiliary supervisions for the table encoder. We test our models on the widely used dataset WIKIBIO which contains Wikipedia infoboxes and related descriptions. The dataset contains complex tables as well as large number of attributes across different domains. We achieve the state-of-the-art performance on both automatic and human evaluation metrics.

Introduction

Data-to-text generation produces understandable texts from some underlying non-linguistic representation of information (Reiter and Dale 1997; 2000). Table-to-text generation, which belongs to the data-to-text generation, aims at generating natural language descriptions for the structured tables to help people to get the key points of the tables.

Different from text-to-text generation tasks like machine translation or abstractive summarization, the sources for table-to-text generation are the tables with hierarchical attribute-value structure. Open-domain tables like Wikipedia infoboxes (Table 1) often have large number of attributes across different domains.

Although previous researchers proposed some task-specific encoder-decoder models for table-to-text generation (Liu et al. 2017a; Sha et al. 2017; Wiseman, Shieber, and Rush 2017; Perez-Beltrachini and Lapata 2018; Bao et al. 2018), most of them are dedicated to improving the decoding phase while generating (Sha et al. 2017; Wiseman, Shieber, and Rush 2017; Perez-Beltrachini and Lapata 2018; Bao et al. 2018). We argue that a single encoder without any auxiliary assistant may not be effective to capture the accurate semantic representation due to the hierarchical structure and vast heterogeneous attributes of the tables.

Infobox:

Attribute	Content
Name	*Edward Merrill Root*
Birthdate	*04 January 1895*
Birthplace	*Baltimore, Maryland, USA*
Deathdate	*26 October 1973*
Deathplace	*Kennebunkport, Maine*
Nationality	*American*
Known for	*anti-communist activities*
Occupation	*educator and poet*
Alma Mater	*Amherst College*
Residence	*Richmond, Indiana*
Article title	*E. Merrill Root*

Description: Edward Merrill Root , known as e. Merrill Root (January 4 , 1895 - October 26 , 1973) , was an American educator and poet devoted to anti-communist causes .

Table 1: Example of a Wikipedia infobox for '*Edward Merrill Root*' and the associated description.

To this end, we first propose a two-level hierarchical table encoder, which encodes both the word-level and attribute-level semantics. The coarse-to-fine attention is proposed to cooperate with the two-level hierarchical table encoder.

Although the hierarchical table encoder has greatly improved the table-to-text generation, we believe the performance can be further enhanced by the external assistance to the table encoder as a complex table may not be well represented by a single encoder. So we also propose 3 auxiliary tasks: including *auxiliary sequence labeling task*, *text autoencoder* and *multi-labeling classification*, to help the table encoder to better represent the source tables and then improve the encoder-decoder style table-to-text generation.

Auxiliary sequence labeling task is a new approach to incorporate the attribute information into the table encoder. We view the attribute names (such as 'Name', 'Birthdate' in Table 1) as the labels for the related table content and use

the sequence labeling as a multi-task to better represent a structured table. This is a way to guide the encoder to reproduce thus 'remember' the attribute names. The experiments show that this is a better way to integrate attribute information than previous works (Lebret, Grangier, and Auli 2016; Sha et al. 2017; Liu et al. 2017a).

Text auto-encoder used the related descriptions of the source tables to supervise the table encoder. Compared with the complex structure of the source tables, the associated descriptions are well-written and straightforward. So it is easier to encode their semantic representation. Since the descriptions share the similar meanings of the associated source tables, it is possible to supervise the learning of the semantic representation of the source tables with that of the related descriptions. We use the internal representation of the auto-encoder to supervise that of the hierarchical table encoder by minimizing their distance.

Multi-labeling classification is also operated on the internal representation of the table encoder. We view all the attribute names which appear in the specific table as the targets for the multi-label classification on the internal representation of the table encoder. We encourage the semantic representation of the source tables to carry as much attribute-level information as possible.

The auxiliary supervisions from *text auto-encoder* and *multi-labeling classification* can be widely applied into all the encoder-decoder models for table-to-text generation. The *auxiliary sequence labeling supervision* can only be used in the proposed hierarchical table encoder. Furthermore, we witness a sharp decrease on the performance of the vanilla encoder-decoder models when we randomly shuffle the order of the attributes in the source tables (in both training and testing set). The proposed *auto-encoder* and *multi-labeling* supervisions can make our models more robust to the disordered tables.

We use WIKIBIO (Lebret, Grangier, and Auli 2016), which contains wikipedia infoboxes and related biographies, as our benchmark dataset. The dataset owns over three thousands attributes which describe people in the different areas (domains), including sportsmen, politicians, artists, soldiers, etc. Each table contains about 20 attributes on average. Experiments show our model achieves the state-of-the-art results on both automatic and human evaluation metrics.

Two-level Hierarchical Table Encoder

Notations

Given a table-to-text dataset with N data samples, the t-th data sample $(T_t,\ Y_t)$ contains a source table $T_t = \{x_1, x_2, \cdots, x_m\}$ with m words, and a description $Y_t = \{y_1, y_2, \cdots, y_L\}$ with L words. The source table T_t also has n attributes $A_t = \{a_1, a_2, \cdots, a_n\}$. Each word x_i belongs to a specific attribute a_j. In the following sections, we use $x_i^{a_j}$ to represent the i-th word x_i in the table T_t which belongs to $a_j (x_i \in a_j)$.

Two-level Hierarchical Encoder

Most previous work (Liu et al. 2017a; Sha et al. 2017; Wiseman, Shieber, and Rush 2017) viewed a structured table as a sequence of words and only used word-level LSTM (Hochreiter and Schmidhuber 1997) to encode the tables. However a factual table is inherently organized in the attribute-value hierarchical structure, thus we propose attribute-level LSTM, apart from word-level LSTM, to capture the attribute-level semantics of the table.

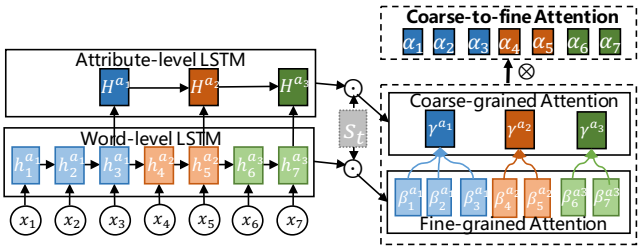


Figure 1: The proposed two-level hierarchical LSTM and coarse-to-fine attention. Suppose we have 3 attributes (marked by 3 different colors) and 7 words in the table. At the t-th decoding step, the decoder state s_t attends to the hidden states of attribute-level and word-level LSTMs for coarse-grained and fine-grained attention, respectively.

As shown in Fig 1, the proposed table encoder contains two separate LSTMs: the word-level LSTM encodes each word $x_i^{a_j}$ in the table sequentially while the attribute-level LSTM encodes each attribute-value tuple by taking the last state $h_{\mathbf{last}}^{a_j}$ for the attribute a_j in the word-level LSTM as input.

$$h_i^{a_j} = \mathbf{LSTM_{word}}(h_{i-1}^{a_k}, x_i^{a_j}) \quad (1)$$

$$H^{a_j} = \mathbf{LSTM_{attribute}}(H^{a_{j-1}}, h_{\mathbf{last}}^{a_j}) \quad (2)$$

We have $a_k = a_j$ if $x_{i-1} \in a_j$, otherwise $a_k = a_{j-1}$.

Coarse-to-fine Attention

To cooperate with the proposed two-level LSTM, we also modify the attention mechanism to incorporate both word-level and attribute-level semantics.

For convenience, we only focus on the one step of decoding in the following illustration. Given the decoder state s_t at the t-th decoding step, $\beta_i^{a_j}$ is the fine-grained attention for the i-th word $x_i^{a_j}$ in the table, which belongs to the attribute a_j. γ^{a_j} is the coarse-grained attention for the attribute a_j.

$$\beta_i^{a_j} \propto g(h_i^{a_j}, s_t); \gamma^{a_j} \propto g(H^{a_j}, s_t) \quad (3)$$

in which $g(\cdot)$ is the Bahdanau-style attention calculation function (Bahdanau, Cho, and Bengio 2014).

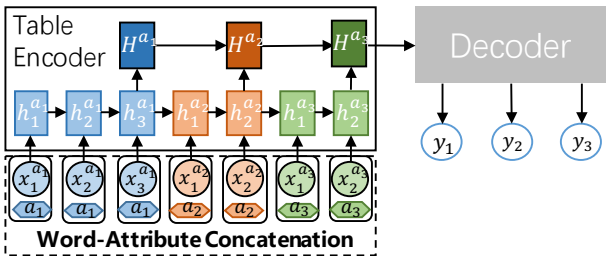
The proposed coarse-to-fine attention α_i for the i-th word in the table is the element-wise product of the fine-grained attention $\beta_i^{a_j}$ and the coarse-grained attention γ^{a_j}.

$$\alpha_i = \beta_i^{a_j} \times \gamma^{a_j}(x_i \in a_j) \quad (4)$$

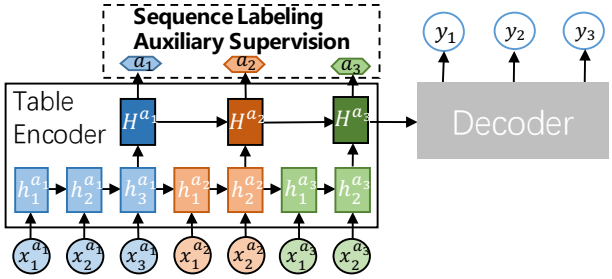
Auxiliary Supervision For Table Encoder

Why we need auxiliary supervision?

1) Many open-domain table-to-text datasets, such as WIKIBIO, have large numbers of attributes across different domains. Even an individual table in WIKIBIO has about 20



(a) Word-Attribute Concatenation



(b) Auxiliary Sequence Labeling Supervision

Figure 2: Two ways to incorporate attribute information into the encoder. Most previous work utilized word-attribute concatenation to represent the tables (Fig a). We find a more efficient way to guide the encoder to 'remember' the attributes (Fig b). We treat the attribute names as the labels (dashed box) for a sequence labeling auxiliary task and train it jointly with the table-to-text generation task.

attributes on average. So it is very challenging for a single encoder to incorporate such heterogeneous resources and learn the accurate semantic representation of the source tables.

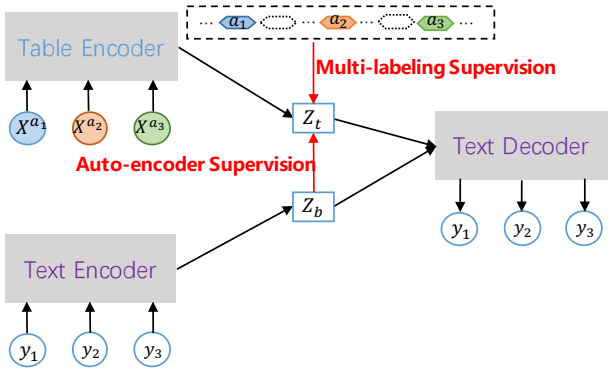
2) Due to the extreme data hungry of the neural network models, researches often use large-scale crawled datasets from the Internet, however, some informal expressions or outdated information may appear as noisy cases in the structured tables.

3) As studied in (Sha et al. 2017), the order of different key-value pairs does influence the generation quality, the attributes feed earlier to the encoder may be ignored due to the gradient vanishing and exploding problem in the encoder-decoder framework.

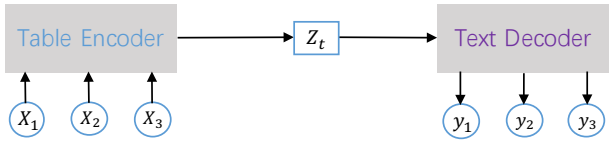
Auxiliary Sequence Labeling Task

Although it is quite straightforward to incorporate the attribute names as the additional inputs to the table encoder (Hachey, Radford, and Chisholm 2017; Liu et al. 2017a; Sha et al. 2017), there is no explicit evidence showing that the encoder can 'remember' the attributes. To this end, we treat the attribute information as an auxiliary supervision to explicitly guide the table encoder to reproduce (thus 'remember') the attribute names in the tables by a sequence-labeling multi-task.

As shown in Fig 2, for a specific table T_k with n attributes, the sequence labeling training is based on the hidden states



(a) Training Stage



(b) Testing Stage

Figure 3: The overview of auto-encoder and multi-labeling supervision. We use text auto-encoder to supervise the table encoder as the text has similar meaning with the source tables. We also use the multi-label classification as auxiliary assistant task . At the training stage, we use the auxiliary tasks to supervise the table-to-text model. At the testing stage, we only use table-to-text model to generate texts.

$[H^{a_i}]_{i=1:n}$ of the attribute-level LSTM.

$$a^*_{1:n} = \mathbf{LABEL}([H^{a_i}]_{i=1:n}) \quad (5)$$

in which $\mathbf{LABEL}$ function can be multi-layer perceptron (MLP) or CRF (Lafferty, McCallum, and Pereira 2001) layer, $a^*_{1:n}$ are the predicted attribute names.

The auxiliary sequence labeling task is trained jointly with the sequence-to-sequence learning. We use cross-entropy loss for the sequence labeling task.

$$L_{SL} = -\lambda_1 \sum_{k=1}^{N} p_{\theta_{SL}}(A_k|\mathbf{H_k}) \quad (6)$$

λ_1 is a tunable hype-parameter which is set to 0.3 according to the validation set. A_k and $\mathbf{H_k}$ are the set of attribute names and the attribute-level hidden states for Table T_k.

Auto-encoder Supervision

Compared with the complex structured tables, their associated descriptions are also strong guidances to the representation of the source tables, as they are well-written and share the similar meanings as the source tables. Therefore, we propose a text auto-encoder to reconstruct the descriptions.

At the training stage, the table encoder compresses the source table T_i into an internal representation z_t. At the same time, the text encoder compresses the reference text Y_i into the representation z_b. Then both z_t and z_b are fed into the same decoder to generate the related description. The loss

of the text auto-encoder is also the cross-entropy losses:

$$L_{AE} = -\lambda_2 \sum_{i=1}^{N} p_{\theta_{AE}}(Y_i|z_b) \quad (7)$$

where λ_2 is a also tunable hyper-parameter, which is set to 0.5 according to the validation set.

We enhance the supervision of text auto-encoder by minimizing the distance between the semantic representation z_t and z_b. We implement the supervision by adding L_{dis} into the loss function.

$$L^{'}_{AE} = L_{AE} + L_{dis}; L_{dis} = \frac{\lambda_3}{N_h} d(z_t, z_b) \quad (8)$$

where $d(z_t, z_b)$ is a function which measures the distance between z_t and z_b. λ_3 is a tunable hyper-parameter to balance the loss of the supervision and other parts of the overall loss. N_h is the number of the hidden unit to limit the magnitude of the distance function. We set $\lambda_3 = 0.3$ based on the model performance on the validation set. the distance between two representations can be written as L2 constraint:

$$d(z_t, z_b) = ||z_t - z_b||_2 \quad (9)$$

Multi-labeling Supervision

As shown in Fig 3, to encourage the internal representation z_t of the table to carry as much attribute-level information as possible, we also use multi-label classification task on z_t to encode the attributes across different domains.

We use MLP to map z_t to the labels $\widetilde{a}^*_{1:n}$ and treat all the attributes $A_i = a_{1:n}$ in the associated table T_i as the targets. λ_4 is also a hyper-parameter which is set to 0.5 according to the validation set.

$$L_{ML} = -\lambda_4 \sum_{i=1}^{N} p_{\theta_{ML}}(A_i|z_t) \quad (10)$$

Loss Function and Training

The overall objective function consists of 4 parts: cross-entropy loss of the table-to-text generation, auxiliary sequence labeling task (Eq 6), text auto-encoder supervision (Eq 7 & 8), and the multi-labeling supervision (Eq 10). The cross-entropy loss of the table-to-text generation is written as:

$$L_{S2S} = -\sum_{i=1}^{N} p_{\theta_{S2S}}(Y_i|z_t) \quad (11)$$

The overall loss of our model is the sum of these individual losses.

$$L = L_{S2S} + L_{SL} + L^{'}_{AE} + L_{ML} \quad (12)$$

Experiments

Dataset

We use WIKIBIO dataset (Lebret, Grangier, and Auli 2016) as our benchmark dataset. WIKIBIO contains 728,321 articles from English Wikipedia (Seq 2015). The dataset uses the first sentence of each article as the description of the related infobox, which contains 26.1 words on average. 9.5 words in the description also occur in the infoboxes. The infobox contains 53.1 words and 19.7 attributes on average. The dataset has been divided in to training (80%), testing (10%) and validation (10%) sets.

Evaluation Metrics

Automatic Metrics: Following the previous work (Lebret, Grangier, and Auli 2016; Sha et al. 2017; Liu et al. 2017a), we use BLEU-4 (Papineni et al. 2002) and ROUGE-4 (F measure) (Lin 2004) for automatic evaluation.

Human Evaluation: Since automatic evaluations like BLEU may not always be reliable for NLG systems. We use human evaluation which involves the generation fluency and the generation quality (how much false or irrelevant information is mentioned in the biography). We firstly sampled 300 generated items from the test set for human evaluation. Each item contains the generated descriptions by different systems given the same resource tables. These items are distributed to 3 third-party crowd-workers who have no knowledge about which system the biography is from. They are asked to score the generated biographies according to their fluency and quality. The scores range from 1 to 5 (higher scores are better). Table 5 shows the scores for the generated biographies whose source table is Table 1.

Experimental Details

Following previous work (Liu et al. 2017a). We select the most frequent 20,000 words in the training set as the word vocabulary. For attribute vocabulary, we select the most frequent 1480 attributes.

We tune the hyper-parameters based on the model performance on the validation set. Since we have many hyper-parameters ($\lambda_1 - \lambda_4$) in Eq 6, Eq 7, Eq 8 and Eq 10. For convenience, We only tune each λ independently from [0.3,0.5,0.7,1.0]. The dimensions of word embedding, field embedding, hidden unit are set as 500, 50, 600 respectively. The batch size, learning rate and optimizer are 32, 3e-4 and Adam (Kingma and Ba 2014), respectively. We use Xavier initialization for all the parameters in our model. We replace UNK tokens with the most relevant token in the source table according to the attention matrix (Jean et al. 2014). The results in Table 2 come from the results of **5 independent runs** of the models.

Baselines

- **KN & Template KN:** The template-based Kneser-Ney (KN) language model reported in (Lebret, Grangier, and Auli 2016). They used the KenLM tool to train a 5-gram models. The extracted template for the biography in Table 1 is "*name_1 name_2 name_3* (*birthdate_2* $\cdots$ " During inference, the decoder is constrained to emit words from the vocabulary or the special tokens occurring in the table.
- **NLM & Table NLM:** Lebret, Grangier, and Auli 2016 proposed a neural language model (NLM) which ignores attribute information. Table NLM includes local and global conditioning on the tables by taking the attribute information into consideration.

Models	BLEU	ROUGE
KN	2.21	0.38
Template KN	19.80	10.70
NLM	4.17	1.48
Table NLM	34.70	25.80
Order-planning	43.91	37.15
Struct-aware	44.89 ± 0.33	41.21 ± 0.25
Our models		
Seq2seq†	43.37 ± 0.32	39.78 ± 0.21
+ Two-level LSTM†	44.14 ± 0.24	40.25 ± 0.15
+ Coarse-to-fine Attention†	44.42 ± 0.21	40.37 ± 0.26
+ Sequence labeling (Eq 6)‡	44.63 ± 0.29	40.64 ± 0.32
+ Auto-encoder (Eq 7 & 8)‡	44.84 ± 0.27	40.91 ± 0.19
+ Multi labeling (Eq 10)‡	**45.14** ± 0.34	**41.26** ± 0.37

Table 2: Automatic evaluation on the WIKIBIO dataset. '+ *A*' means this model adds module *A* to the last model. All the baselines are reported by their authors. Models marked by † use word-attribute concatenation (Fig 2 (a)) to incorporate attribute information while models marked by ‡ use auxiliary sequence labeling supervision (Fig 2 (b)).

Models	Fluency	Quality
seq2seq	4.23	2.89
Struct-aware (Liu et al. 2017a)	4.42	3.64
Our best	**4.56**	**3.85**

Table 3: Human evaluation (larger values signifies better performances) for 3 systems. The generated cases of the state-of-the-art system (Liu et al. 2017a) are provided by the authors. The Pearson coefficients of the annotators' scores on the generation fluency and quality are **0.77** and **0.68** respectively (both p-values less than 0.001).

- **Order-planning:** Sha et al. 2017 proposed a link matrix to model the order for the attribute-value tuples while generating the related biography.
- **Struct-aware:** Liu et al. 2017a proposed a structure-aware learning which include the 'field-gating' mechanism which input the attribute name embedding into the LSTM and the dual attention mechanism to incorporate the attribute information.

Analysis for Human & Automatic Evaluation

The automatic evaluation (Table 2) shows that our proposed model outperform the seq2seq baseline by about 1.8 BLEU and 1.5 ROUGE and also beat the state-of-the-art system (Liu et al. 2017a). The proposed two-level LSTM encoder with coarse-to-fine attention brings 1.1 BLEU and 0.6 ROUGE increase compared with the vanilla seq2seq model. Table 4 shows that the improvement of the two-level LSTM structure comes from better representation of the attribute-value structure of the tables rather then increased parameters. The auxiliary supervisions from different resources further enhance the model performance. Human evaluations in Table 3 also shows that the generated cases from our model have higher quality than the seq2seq baseline and the state-of-the-art system.

Encoder	BLEU	ROUGE
Two-level LSTM (Ours)	**44.14**	**40.25**
One-layer LSTM	43.37	39.78
Two-layer LSTM	43.25	39.53
Bi-directional LSTM	43.41	39.64

Table 4: The comparison of the proposed two-level LSTM encoder with stacked LSTM encoders. It shows that the improvement of the two-level hierarchical encoder does not come from the increased model parameters .

s2s: Edward Merrill Root (January 4 , 1895 – October 26 , 1973) was an american educator and poet .
Liu et al. 2017a: Edward Merrill Root (January 4 , 1895 – October 26 , 1973) was an american educator and poet from Richmond , Indiana .
Our best: Edward " E. " Merrill Root (January 4 , 1895 in Baltimore – October 26 , 1973 in Kennebunkpot) was an american educator and poet , best known for his anti-communist activities .

Table 5: The generated biographies and associated human evaluation scores for Table 1 from 3 systems. All the biographies are scored **5.0** for the *fluency* evaluation. The average scores of generation *quality* for the 3 systems are **2.67**, **3.67** and **4.0** respectively.

Ablation Studies

We first introduce the baselines in Table 6 and 7:

- **hs2s + concat**: The proposed hierarchical encoder with two-level LSTM and coarse-to-fine attention (Fig 1) using word-attribute concatenation (Fig 2(a)).
- **hs2s + gating**: The proposed hierarchical encoder with field-gating mechanism (Liu et al. 2017a). We feed the attribute name embedding into both the word-level and

Models	BLEU
hs2s + concat	44.42
hs2s + gating	44.57
hs2s + SL-MLP	**44.63**
hs2s + SL-CRF	44.54

(a) Different Ways to incorporate attribute information

Models	ACC(%)
hLSTM + MLP	94.01
hLSTM + CRF	**94.30**
hs2s + SL-MLP	93.84
hs2s + SL-CRF	94.07

(b) The accuracy of sequence labeling

Table 6: The analysis of auxiliary sequence labeling supervision. We show that the proposed sequence labeling auxiliary task performed better than word-attribute concatenation (Fig 2 (a)) and attribute-gating method (Liu et al. 2017a) (left). The accuracies of the sequence labeling multi-task are comparable with competitive baselines (right).

Models	BLEU	ROUGE
s2s	43.37	39.78
+ Co-training (Eq 7)	43.69(+0.32)	40.02(+0.24)
+ Dis. Minimizing (Eq 8)	44.04(+0.67)	40.22(+0.44)
hs2s + concat	44.42	40.37
+ Co-training (Eq 7)	44.64(+0.22)	40.68(+0.31)
+ Dis. Minimizing (Eq 8)	44.68(+0.26)	40.74(+0.37)
hs2s + SL-CRF	44.63	40.64
+ Co-training (Eq 7)	44.76(+0.13)	40.86(+0.22)
+ Dis. Minimizing (Eq 8)	44.84(+0.21)	40.91+(0.27)

(a) Ablation studies on Auto-encoder Supervision

Models	BLEU	ROUGE
s2s	43.37	39.78
+ Multi-labeling (Eq 10)	43.69(+0.32)	40.02(+0.24)
hs2s + concat	44.42	40.37
+ Multi-labeling (Eq 10)	44.77(+0.35)	40.63(+0.26)
hs2s + SL-CRF	44.63	40.64
+ Multi-labeling (Eq 10)	45.01(+0.38)	41.06(+0.42)

(b) Ablation studies on Multi-labeling Supervision

Table 7: Ablation studies on the auto-encoder and multi-labeling supervision.

attribute-level LSTM units. Please refer to (Liu et al. 2017a) for more details.

- **hs2s + SL-MLP/CRF**: The proposed hierarchical encoder which integrates attribute-level information by joint training the auxiliary sequence labeling task with the seq2seq learning. The $\mathbf{LABEL}$ function in Eq 5 can be MLP or CRF.
- **hLSTM + MLP/CRF**: The independent sequence labeling model using the two-level hierarchical LSTM. We can also use MLP or CRF as the $\mathbf{LABEL}$ function.

Table 6 (a) shows that the proposed sequence labeling auxiliary task is a better way to incorporate attribute information than word-attribute concatenation (Fig 2(a)) and the field-gating mechanism (Liu et al. 2017a). Table 6 (b), on the other hand, shows the accuracy of the sequence labeling auxiliary task on the test set of WIKIBIO. The joint training models (hs2s + SL-MLP/CRF) also achieve comparable sequence labeling accuracies with the independent sequence labeling models (hLSTM + MLP/CRF).

The sequence labeling auxiliary task can only be operated on the proposed two-level LSTM. The auto-encoder and multi-labeling supervision, on the other hand, can be applied to all seq2seq-like models. Table 7 shows that both auto-encoder and multi-labeling supervision can be helpful to encode the structured tables and improve the table-to-text generation.

We notice more increase on the automatic evaluations on the vanilla seq2seq model with the help of the auto-encoder supervision (Table 7 (a)). We believe this is because the less expressive encoder might benefit more from the auxiliary auto-encoder supervision from the related text. For the multi-labeling supervision, all the baselines can get about 0.3 increase on both BLEU and ROUGE metrics.

Key	Value
Name	*Bobby Fenwick*
Position	*infielder*
Birthdate	*10 December, 1946*
Birthplace	*Naha, Okinawa*
Debutdate	*April 26, 1972*
Debutteam	*Houston Astros*
Finaldate	*May 8, 1973*
Finalteam	*Louis Cardinals*

Gold: Robert Richard Fenwick (December 10 , 1946 in Naha , Okinawa) , is a retired major league baseball player who played infielder from 1972 - 1973 .
s2s: Robert Joseph Fenwick (born December 10 , 1946 in Naha , Okinawa) is a former major league baseball infielder .
+ hierarchical encoder: Robert Fenwick (born December 10 , 1946 in Naha , Okinawa) is a former major league baseball infielder who played for Houston Astros and Louis Cardinals.
+ Auxiliary Supervision: Robert Fenwick (born December 10 , 1946 in Naha , Okinawa) is a former major league baseball player who played as infielder for Houston Astros and Louis Cardinals from 1972 to 1973 .

Table 8: The generated biographies for '*Bobby Fenwick*' in the WIKIBIO.

Generation Analysis

We offer 2 generated cases (Table 5 and Table 8) for the infoboxes in the test set of WIKIBIO. Table 5 shows that the generated biography by our model get higher human evaluation scores than its counterpart provided by the state-of-the-art system (Liu et al. 2017a). In this case, our system includes the information in the 'known for' attribute. Similarly, the generated biography by our system in Table 8 contains the teams where *Bobby Fenwick* played for as well as the time span in these teams.

Disordered Tables

Most previous work (Lebret, Grangier, and Auli 2016; Liu et al. 2017a; Sha et al. 2017) viewed the attributes in the source tables as a ordered list and then feed them sequentially into the table encoder. Actually, as proved by (Sha et al. 2017; Liu et al. 2017a), the order of the attributes does influence the generation quality. The conclusion makes perfect sense, especially for Wikipedia infoboxes and associated biographies, as the human editors tend to describe a person in a relatively fixed order. For example, most biographies accords with the 'name-birthdate-nationality-occupation-$\cdots$' pattern. The attributes in the infoboxes are usually arranged in the proper order.

However, not every table is guaranteed to have the appropriate order. A robust model should achieve constantly

Models	BLEU
s2s	41.80 (-1.57)
s2s + Auto-encoder (Eq 7 & 8)	43.19 (-0.85)
s2s + Multi-labeling (Eq 10)	43.06 (-0.63)
s2s + AE & ML (Eq 7, 8 & 10)	**43.89 (-0.52)**
hs2s	43.21 (-1.21)
hs2s + SL-CRF (Eq 5)	43.46 (-1.17)
hs2s + Auto-encoder (Eq 7 & 8)	44.04 (-0.64)
hs2s + Multi-labeling (Eq 10)	44.30 (-0.47)
hs2s + AE & ML (Eq 7, 8 & 10)	**44.51 (-0.43)**
hs2s + AE & ML & SL-CRF (full)	44.58 (-0.56)

Table 9: The performance of different models on the disordered tables, which shows that the text auto-encoder and multi-labeling supervisions can make the models more robust to the disordered tables.

high performance even if the attributes are disordered. So we test our model in the adversarial setting. We randomly shuffle the attributes of the source tables in both training and testing set. During the shuffling, we only reorder the attributes of the source tables without changing the content in these attributes. We shuffle for 3 times and average the scores of the candidate models on the shuffled data sets during this process. Table 9 shows that the shuffling does hurt the performance of our models. However, we find that the proposed auto-encoder and multi-labeling auxiliary supervision can relieve the undesirable tendencies. We believe this is because the two sources of external supervision can facilitate the table encoder to learn more accurate semantic representations for the source tables no matter how the attributes are ordered.

Error Analysis

Although our models have greatly improved the table-to-text generation, we also find some errors by case studies. 1) The first problem is the irrelevant information in the generated descriptions to the source tables. it is a general problem in the seq2seq framework as we usually view the seq2seq models as the black boxes and can not easily debug the models according to the ill-formed generations. 2) In some cases, we need common sense knowledge to get the reasonable biographies. For example, when we say 'a retired basketball player', we should determine whether a man is retired or not according to the 'Finaldate' attribute. 3) The information which needs some inference across several attributes (like a time span) may not be well represented by our model.

Related Work

Auto-encoder has been shown to be effective to learn the internal representations (Vincent et al. 2008) for the source data in various domains, including natural language understanding (AP et al. 2014), speech recognition (Deng et al. 2010; Lu et al. 2013) and image representation (Krizhevsky and Hinton 2011). Our model utilizes a biography auto-encoder to supervise the learning for the related table representation. A closely related work is the auto-encoder assistant proposed by Ma et al. 2018 which used the text auto-encoder in the field of abstractive summarization.

Sequence labeling, which labels the source sequences with pre-defined labels. Hand-crafted domain-specific features were widely used in traditional methods, like HMMs and CRFs (Lafferty, McCallum, and Pereira 2001; McCallum and Li 2003). Recently, there are many attempts to build end-to-end systems for sequence labeling (Lample et al. 2016; Ma and Hovy 2016). Our model is based on the success of LSTM+MLP and LSTM+CRF models.

Natural language generation tasks can be generally divided into two phases: *content selection* ('what to say') and *surface realization* ('how to say')(Reiter and Dale 1997; 2000). Many previous work (Barzilay and Lee 2004; Barzilay and Lapata 2005; 2006; Yu et al. 2007; Liang, Jordan, and Klein 2009) treated the task as a pipelined systems, which viewed content selection and surface realization as two separate tasks. Duboue and McKeown (2002) proposed a clustering approach in the biography domain by scoring the semantic relevance of the text and paired knowledge base. In a similar vein, Barzilay and Lapata (2005) modeled the dependencies between the American football records and identified the bits of information to be verbalized. (Liang, Jordan, and Klein 2009; Angeli, Liang, and Klein 2010) extend the work of Barzilay and Lapata to soccer and weather domains by learning the alignment between data and text using hidden variable models. Most recent work treated natural language generation in an end-to-end fashion (Mei, Bansal, and Walter 2016; Lebret, Grangier, and Auli 2016; Wiseman, Shieber, and Rush 2017; Xu et al. 2018; Lin et al. 2018; Luo et al. 2018; Liu et al. 2017b; Wang et al. 2017) with the help of attention mechanism (Bahdanau, Cho, and Bengio 2014; Luong, Pham, and Manning 2015; Luo et al. 2018; Wu et al. 2018).

Conclusion

Many tables have complex attribute-value hierarchical structure and large number of attributes across different domains. So it is hard for a single encoder to learn the accurate semantic representation of the source tables. To this end, we first propose a two-level hierarchical encoder with coarse-to-fine attention to encode the tables. Furthermore, we also propose 3 auxiliary tasks to assist the table encoder, namely *auxiliary sequence labeling task*, *text auto-encoder* and *multi-label classification*. The experiments on the `WIKIBIO` dataset show that our models achieve the state-of-the-art performance and are more robust in the adversarial setting.

Acknowledgments

We would like to thank Xiaodong Li, Wei Wu and Kexiang Wang, as well as the anonymous reviewers for the helpful discussions and suggestions. Our work is supported by the National Science Foundation of China under Grant No. 61876004, No. 61751201 and No. M1752013. The corresponding authors of this paper are Baobao Chang and Zhifang Sui.

References

Angeli, G.; Liang, P.; and Klein, D. 2010. A simple domain-independent probabilistic approach to generation. In *EMNLP 2010*, 502–512.

AP, S. C.; Lauly, S.; Larochelle, H.; Khapra, M.; Ravindran, B.; Raykar, V. C.; and Saha, A. 2014. An autoencoder approach to learning bilingual word representations. In *NIPS*, 1853–1861.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

Bao, J.; Tang, D.; Duan, N.; Yan, Z.; Lv, Y.; Zhou, M.; and Zhao, T. 2018. Table-to-text: Describing table region with natural language. *arXiv preprint arXiv:1805.11234*.

Barzilay, R., and Lapata, M. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 331–338.

Barzilay, R., and Lapata, M. 2006. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*, 359–366.

Barzilay, R., and Lee, L. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. *arXiv preprint cs/0405039*.

Deng, L.; Seltzer, M. L.; Yu, D.; Acero, A.; Mohamed, A.-r.; and Hinton, G. 2010. Binary coding of speech spectrograms using a deep auto-encoder. In *Eleventh Annual Conference of the International Speech Communication Association*.

Duboue, P. A., and McKeown, K. R. 2002. Content planner construction via evolutionary algorithms and a corpus-based fitness function. In *Proceedings of INLG 2002*, 89–96.

Hachey, B.; Radford, W.; and Chisholm, A. 2017. Learning to generate one-sentence biographies from wikidata. In *EACL 2017*, 633–642.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., and Hinton, G. E. 2011. Using very deep autoencoders for content-based image retrieval. In *ESANN*.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lebret, R.; Grangier, D.; and Auli, M. 2016. Neural text generation from structured data with application to the biography domain. In *EMNLP 2016*, 1203–1213.

Liang, P.; Jordan, M. I.; and Klein, D. 2009. Learning semantic correspondences with less supervision. In *ACL*, 91–99.

Lin, J.; Sun, X.; Ren, X.; Li, M.; and Su, Q. 2018. Learning when to concentrate or divert attention: Self-adaptive attention temperature for neural machine translation. *arXiv preprint arXiv:1808.07374*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Liu, T.; Wang, K.; Sha, L.; Chang, B.; and Sui, Z. 2017a. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724*.

Liu, T.; Wei, B.; Chang, B.; and Sui, Z. 2017b. Large-scale simple question generation by template-based seq2seq learning. In *National CCF Conference on Natural Language Processing and Chinese Computing*, 75–87. Springer.

Lu, X.; Tsao, Y.; Matsuda, S.; and Hori, C. 2013. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, 436–440.

Luo, L.; Xu, J.; Lin, J.; Zeng, Q.; and Sun, X. 2018. An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. *arXiv preprint arXiv:1808.08795*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Ma, S.; Sun, X.; Lin, J.; and Wang, H. 2018. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. *arXiv preprint arXiv:1805.04869*.

McCallum, A., and Li, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL 2003*, 188–191.

Mei, H.; Bansal, M.; and Walter, M. R. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *NAACL HLT 2016*, 720–730.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL2002*, 311–318.

Perez-Beltrachini, L., and Lapata, M. 2018. Bootstrapping generators from noisy data. *arXiv preprint arXiv:1804.06385*.

Reiter, E., and Dale, R. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3(1):57–87.

Reiter, E., and Dale, R. 2000. *Building natural language generation systems*. Cambridge university press.

Sha, L.; Mou, L.; Liu, T.; Poupart, P.; Li, S.; Chang, B.; and Sui, Z. 2017. Order-planning neural text generation from structured data. *arXiv preprint arXiv:1709.00155*.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, 1096–1103. ACM.

Wang, K.; Liu, T.; Sui, Z.; and Chang, B. 2017. Affinity-preserving random walk for multi-document summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 210–220.

Wiseman, S.; Shieber, S. M.; and Rush, A. M. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Wu, W.; Wang, H.; Liu, T.; and Ma, S. 2018. Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3729–3738.

Xu, J.; Zhang, Y.; Zeng, Q.; Ren, X.; Cai, X.; and Sun, X. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. *arXiv preprint arXiv:1808.06945*.

Yu, J.; Reiter, E.; Hunter, J.; and Mellish, C. 2007. Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering* 13(1):25–49.