Improving Encoder by Auxiliary Supervision Tasks for Table-to-Text Generation

Liang Li[1,2], **Can Ma**[1]*, **Yinliang Yue**[1]* **and Dayong Hu**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Heilongjiang Network Space Research Center, Harbin 150010, China
{liliang, macan, yueyinliang}@iie.ac.cn
superhudayong@163.com

Abstract

Table-to-text generation aims at automatically generating natural text to help people conveniently obtain salient information in tables. Although neural models for table-to-text have achieved remarkable progress, some problems are still overlooked. Previous methods cannot deduce the factual results from the entity's (player or team) performance and the relations between entities. To solve this issue, we first build an entity graph from the input tables and introduce a reasoning module to perform reasoning on the graph. Moreover, there are different relations (e.g., the numeric size relation and the importance relation) between records in different dimensions. And these relations may contribute to the data-to-text generation. However, it is hard for a vanilla encoder to capture these. Consequently, we propose to utilize two auxiliary tasks, Number Ranking (NR) and Importance Ranking (IR), to supervise the encoder to capture the different relations. Experimental results on ROTOWIRE and RW-FG show that our method not only has a good generalization but also outperforms previous methods on several metrics: BLEU, Content Selection, Content Ordering.

1 Introduction

Table-to-text generation is an essential task for text generation from structured data. It aims at automatically producing descriptive natural language text to help people obtain the salient information from the tables. Over the past several years, neural text generation methods have made significant progress on this task. Lebret et al. (2016); Wiseman et al. (2017); Bao et al. (2018) view the input table as a record sequence and model it as a machine translation task. To generate text containing more salient and well-organized facts, Sha et al. (2018); Moryossef et al. (2019); Trisedya et al. (2020); Bai et al. (2020) explicitly model content selection and planning. To better represent tables, Liu et al. (2018); Nema et al. (2018); Gong et al. (2019) explicitly model the structure of a table from multiple levels or different dimensions.

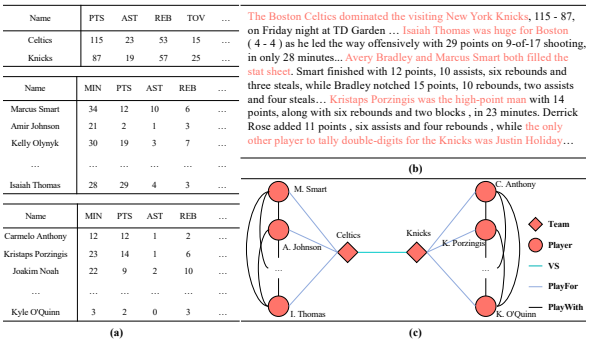

Figure 1: (a) are tables in ROTOWIRE. (b) is a human-written summary related to (a). Factual results that need be reasoned are in red. (c) is the entity graph constructing from the input tables.

Figure 1 (a) contains basketball game statistical tables from ROTOWIRE (Wiseman et al., 2017), a benchmark of NBA basketball games. As can be seen, each entity (player or team) takes one row in the corresponding table. Moreover, each row comprises several records of different types, which describe the entity's performance in different aspects. In terms of generating a summary from these tables, it is necessary to make reasoning to obtain some factual results from the entities' performance and the relationships between entities. For instance, when humans describe the tables in Figure 1 (a), they usually give some factual results, such as "The Boston Celtics dominated the visiting New York Knicks" or "Isaiah Thomas was huge for Boston...". These results need to be reasoned from the entities' performance and the relationships between entities. Therefore, it is necessary to give the model the reasoning ability. However, previous methods do not explicitly model this ability.

*Corresponding authors: Can Ma, Yinliang Yue

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5979–5989
August 1–6, 2021.

Numerical tables mean most records in these tables are numerical and are very common. For instance, 86.82% of the records and almost 86.49% of the column types are numeric in ROTOWIRE. We observe that there are different relations between records in different dimensions. For example, there are two kinds of relations in numerical tables. The first one is numerical size relation in the column dimension, i.e., in the same type column. The other is the relative importance relation in the row dimension. It refers to the relative importance of different types of records, which are in the same row, to the entity that they belong to. On the one hand, these relations may contribute to table-to-text generation. Let us take Figure 1 (a) as an example. I.Thomas's score is 29, which is higher than other records in the column PTS. And he has three rebounds, which is lower than most other records in the column REB. Therefore, humans are more likely to describe his scores rather than his rebounds when summarizing his performance. On the other hand, a vanilla encoder may not effectively capture the relations existing in different dimensions without any auxiliary supervision.

We employ a hierarchical encoder, which comprises a Record Encoder and a Reasoning Module, to encode the input tables from record level and row level. Specifically, inspired by Gong et al. (2019), the Record Encoder utilizes two cascaded self-attention modules to encode the table from the column and the row dimension, respectively. Moreover, to endow the model with the reasoning ability, we first build an entity graph on the row level according to the relations between players and teams. And then, we introduce a reasoning module to perform reasoning on the graph. Furthermore, we utilize different auxiliary tasks to help the encoder capture the different relations among records. More specifically, two auxiliary tasks named Number Ranking (NR) and Importance Ranking (IR) are proposed to supervise the learning of the different parts of the Record Encoder, respectively.

We conducted experiments on ROTOWIRE and RW-FG(Wang, 2019) to verify the effectiveness of the proposed approach. The experimental results demonstrate that it is necessary to enable the model the reasoning ability. Moreover, the proposed two auxiliary tasks can improve the data-to-text model's performance without introducing extra parameters. Furthermore, the results also show our method not only has a good generalization but also outperforms previous methods on BLEU, Content Selection, and Content Ordering metrics.

2 Related Work

Recently, neural models have been the mainstream for table-to-text generation and obtained impressive results. Early works on table-to-text generation regard it as a distinct machine translation task and view a structured table as a record sequence (Lebret et al., 2016; Wiseman et al., 2017; Bao et al., 2018). Most recent works are inspired by the traditional methods for data-to-text generation and introduce explicit content selection and planning to improve the results (Sha et al., 2018; Puduppully et al., 2019b; Moryossef et al., 2019; Trisedya et al., 2020; Bai et al., 2020), and they obtain training labels by aligning the input tables with related summaries. However, this alignment may introduce additional errors. Some works attempt to use additional knowledge to improve the quality of the generated text. Nie et al. (2018) utilize pre-executed symbolic operations on the input table in a sequence-to-sequence model to improve the fidelity of neural table-to-text generation. Chen et al. (2019) introduce the background knowledge of the entity in the table to improve results.

In addition to introducing external knowledge, some works learn better representation for the table by explicitly modeling the table's structure. Liu et al. (2018) propose a structure-aware seq2seq architecture, which incorporates the filed information as the additional inputs to the table encoder. Some works (Bao et al., 2018; Nema et al., 2018; Jain et al., 2018) model the table's representation from the row and column levels, and utilize the dual attention decoder to generate text. Gong et al. (2019) introduce the historical data for each table and utilize a self-attention-based hierarchical encoder on three dimensions (row, column, and time) to enrich the table's representation. Furthermore, Liu et al. (2019) propose three auxiliary supervision tasks (sequence labeling, text auto-encoding, and multi-label classification) to help the encoder capture a more accurate semantic representation of the tables.

Gong et al. (2020) also explicitly model the relations between the numeric records. They pretrain a multi-layer transformer encoder to obtain records' contextual numerical value representations. Moreover, when training the data-to-text model, they replace the record's token embedding with its con-

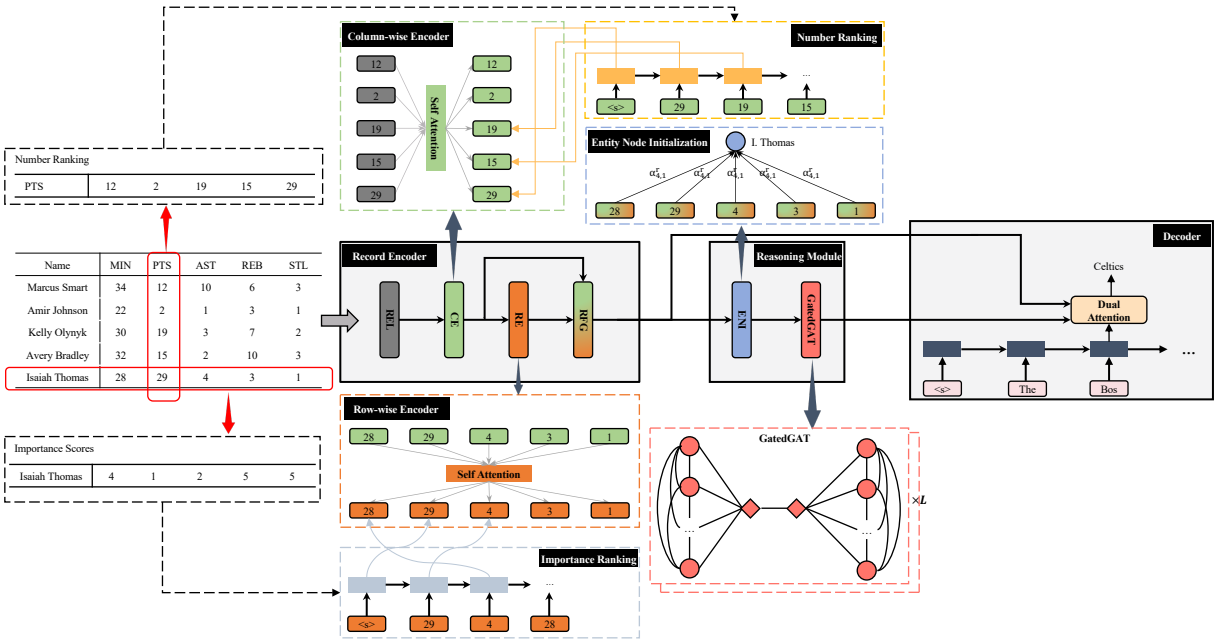

Figure 2: An overview of our method. REL and RFG denote Record Embedding Layer and Record Fusion Gate, respectively.

textual representation from the pre-trained model. Differently, our Number Ranking task is trained with the data-to-text model and can supervise the model actively to capture the numeric size relation without introducing extra parameters.

3 Approach

3.1 Record Encoder

Each input instance consists of three different tables T^1, T^2, T^3, containing records about players' performance in the home team, players' performance in the visiting team, and the team's overall performance. Each cell in the table is regarded as a record. Inspired by Gong et al. (2019), we utilize two self-attention modules to model each record's contexts from the column and the row dimension, respectively. After that, we obtain the fusion representation for records by the record fusion gate.

Record Embedding Following previous work (Wiseman et al., 2017), we utilize four tuples to represent each record r. The four tuples include: entity $r.e$ (the name of team or player, such as Carmelo Anthony), type $r.t$ (e.g., PTS) and value $r.v$ as well as feature $r.f$ (e.g., home or visiting) which indicates whether a player or a team compete in home court or not. And we utilize 1-layer MLP to encode the embeddings of each record's four types of information into a dense vector $r^{emb}_{i,j}$, $r^{emb}_{i,j} = Relu(W^e[r_{i,j}.e; r_{i,j}.t; r_{i,j}.v; r_{i,j}.f] + b^e)$, where i, j denote a record in the table of i-th row and j-th column, $[;]$ denotes the vector concatenation, W^e and b^e are trainable parameters.

Column-wise Encoder To capture the numeric size relation between records, we adopt a self-attention module to model record in the context of other records in the same column and obtain the column dimension representation vector $r^{col}_{i,j}$ as:

$$\alpha^{col}_{i,j,i'} \propto \exp(W^{col}_2 \tanh(W^{col}_1[r^{emb}_{i,j}; r^{emb}_{i',j}]) \quad (1)$$

$$\tilde{r}^{col}_{i,j} = \sum_{i'=1,i'\neq i}^{R} \alpha^{col}_{i,j,i'} r^{emb}_{i',j} \quad (2)$$

$$r^{col}_{i,j} = W^{col}_3[\tilde{r}^{col}_{i,j}; r^{emb}_{i,j}] \quad (3)$$

where W^{col}_1, W^{col}_2 and W^{col}_3 are trainable parameters, R represents the number of rows in the table.

Row-wise Encoder Considering the size relation captured by the Column-wise Encoder (CE) may help the learning of importance relation on row level, we have the Column-wise Encoder and the Row-wise Encoder (RE) in series (as shown in Figure 2). In other words, the input of RE is $r^{col}_{i,j}$ rather than $r^{emb}_{i,j}$. We use another self-attention module, similar to the CE, to obtain the row dimension representation $r^{row}_{i,j}$ for records.

Record Fusion Gate The record representations from different dimensions contribute differently in

reflecting the record's information. Therefore, we utilize a fusion gate to combine the two dimension representations adaptively(Gong et al., 2019). First, we concatenate the two dimension representations of a record and utilize an MLP to obtain a general representation for it as $r^{gen}_{i,j}$. Then, we compare the column dimension representation with $r^{gen}_{i,j}$ to obtain its important score:

$$s^{col}_{i,j} \propto \exp(W^f_2 \tanh(W^f_1[r^{gen}_i; r^{col}_{i,j}])) \quad (4)$$

where W^f_1 and W^f_2 are trainable parameters. Equally, we obtain the important score $s^{row}_{i,j}$ for the row dimension representation $r^{row}_{i,j}$. Finally, we obtain the fused record representation $r^f_{i,j}$ by weighted sum $s^{col}_{i,j} r^{col}_{i,j} + s^{row}_{i,j} r^{row}_{i,j}$. The fused record representations $\{r^f_{i,j}\}^{R,C}_{i=1,j=1}$ will be used as the input of the text decoder.

3.2 Reasoning Module

As mentioned in Section 1, we observe some factual results in text that require reasoning from the entities' performance and the relationships between them. Therefore, it is necessary to enable model the reasoning ability. To achieve this, we primarily build an entity graph according to the entities' relationships in input tables, as shown in Figure 1 (c). And then, we leverage Graph Neural Networks (GNN) to perform reasoning. Following, we describe the details of the reasoning process.

Primarily, we obtain the initialized representation for each entity in tables by the Entity Node Initialization module (ENI). Considering that different records in the same row may not contribute the same, we combine them dynamically by attention mechanism. We first compute a general representation vector e^{gen}_i for the entity e_i, which is given by mean-pooling over the same row records $r^f_{i,1}, r^f_{i,2}, ..., r^f_{i,C}$. Then we compare each record in the i-th row with e^{gen}_i and obtain the initialized entity representation e^0_i by weighted sum:

$$\alpha^r_{i,j} \propto \exp(W^r_2 \tanh(W^r_1[e^{gen}_i; r^f_{i,j}])) \quad (5)$$

$$e^0_i = \sum_{j=1}^{j=C} \alpha^r_{i,j} r^f_{i,j} \quad (6)$$

After obtaining the initial representations of entities, we adopt graph neural networks to propagate entity node information to their neighbors. Inspired by GAT(Velickovic et al., 2018), we use multi-head attention to measure the relatedness between target entity node e_i and its neighbor nodes at layer l:

$$\alpha^l_{i,j} = MultiHeadAttention(e^{l-1}_i, e^{l-1}_j) \quad (7)$$

where $j \in \mathcal{N}_i$ and $\mathcal{N}_i$ means the neighbor nodes set of target entity e_i.

The neighbor entities include information that is not relevant to the target entity. Therefore, we modify the way the information flow in GAT. Explicitly, we incorporate gate mechanisms into information aggregation to filter out noises from neighbor nodes and extract useful information, which we name GatedGAT. The representation e^l_i of e_i at layer l is calculated as follows:

$$e^l_i = gate^l_i * e^{l-1}_i + (1 - gate^l_i) * \tilde{e}^l_i \quad (8)$$

$$\tilde{e}^l_i = ELU(\sum_{j \in N_i} \alpha^l_{i,j} e^{l-1}_j) \quad (9)$$

$$gate^l_i = sigmoid(W^l[e^{l-1}_i; \tilde{e}^l_i]) \quad (10)$$

where W^l is a learnable parameter. The entities' representations $\{e^L_i\}^R_{i=1}$ at the last layer L are employed in text decoder.

3.3 Decoder with Dual Attention

To make use of record-level and row-level semantics information, we adopt the dual attention mechanism. Specifically, at decoding step t, the input of the LSTM unit is the embedding of the previously predicted word y_{t-1}. And given the decoder state d_t, we first calculate the row-level attention $\beta_{t,i}$, which is based on the similarity between the decoder state d_t and the entities' representations $\{e^L_i\}^R_{i=1}$. Then we compute the record-level attention $\alpha_{t,i}$ over all the record representations $\{r^f_{i,j}\}^{R,C}_{i,j}$ which are normalized among records in the same row. Finally, we fuse these two-level attention and obtain the context representation as:

$$\alpha'_{t,i,j} = \alpha_{t,i} \beta_{t,i,j} \quad (11)$$

$$c^d_t = \sum_{i=1}^{R} \sum_{j=1}^{C} \alpha'_{t,i,j} r_{i,j} \quad (12)$$

Given a reference output $\{y_i\}^T_{i=1}$, we use the cross-entropy loss as the objective function of table-to-text generation:

$$L_{lm} = -\sum_{i=1}^{T} p_\theta(y_t | y_{1:t-1}; c^d_t) \quad (13)$$

3.4 Auxiliary Supervision Task

Liu et al. (2019) have shown that a single encoder without any auxiliary assistant may not be effective to capture the accurate semantic representation. Inspired by this, we propose two auxiliary tasks, Number Ranking (NR) and Importance Ranking (IR), to help the Column-wise Encoder and the Row-wise Encoder capture the size relation and the relative importance relation among records respectively.

Number Ranking In practice, many tables mainly comprise numeric records. Different from text-type content, the numerical content contains less semantic information but the size relation. The size relation means the value of a record is larger or smaller than others, and it plays an essential role in records selection. For example, humans tend to focus on the highest scores or the fewest faults in a basketball game table. Therefore, it is necessary to incorporate size relation into record representation. To achieve this, we propose an auxiliary supervision task named Number Ranking (NR) to supervise the learning of the Column-wise Encoder. As shown in Figure 2 top, we take a list of records in column PTS to illustrate how it works. Specifically, we regard the PTS column of the table as an out-of-order set of records $C = r_1, r_2, ..., r_R$, and the goal is to generate a sequence of record pointers in descending order according to their value. We adopt the Pointer Networks (Vinyals et al., 2015) to solve this problem and the output of Column-wise Encoder r_i^{col} (we omitted the indices on the column dimension) as its input. Let $z = z_1, ..., z_R$ denote the sequence of the ranked records' indices. Each z_k points to an input record and is between 1 and R. As shown in Figure 2, we use an LSTM as the decoder. The $MeanPooling(\{r_i\}_{i=1}^R)$ is used as the initialization of the first hidden state of the decoder. At each decoding step t, we calculate a distribution over the input records:

$$h_t = LSTM(h_{t-1}, r_{z_{t-1}}^{col}) \tag{14}$$

$$p_{t,i}^n \propto exp(W_{nr}[h_t; r_i^{col}]) \tag{15}$$

where W_{nr} is a trainable parameter, and $p_{t,i}^n$ denotes the probability that the output points to the record r_i at step t. We take the cross-entropy loss for this task:

$$L_{nr} = -\sum_{j=1}^{C}\sum_{i=1}^{R} \log p_{i,z_i}^n \tag{16}$$

Importance Ranking When people describe a player's performance in a basketball game, they tend to focus on his relatively important record and describe these firstly. Consequently, we introduce the Importance Ranking task (IR) to supervise the Row-wise Encoder to capture the relative importance relations between records in the same row. This task's input is a sequence record in the same row, and the output is a sequence of records in descending order of the records' importance. We employ a pointer network similar to the one used in the Number Ranking task to model this task. However, different from the records in the same column, these in the same row cannot be directly compared as they represent different meanings. To address this issue, we take the rank of each record in the column as an importance indicator. Figure 2 left bottom shows an example of calculating the importance scores for records in the last row of the table.

The input of the decoder is the output of the Row-wise Encoder $\{r_j^{row}\}_{j=1}^R$. And the output is the ascending order of the input, according to the records' importance scores. Let $p_{t,j}^s$ denote the probability of pointing to record r_j at decoding step t, the loss function for this task is:

$$L_{ir} = -\sum_{i=1}^{R}\sum_{j=1}^{C} \log p_{j,z_j}^s \tag{17}$$

3.5 Loss Function and Training

These two tasks are trained together with the table-to-text task, and the overall objective function consists of three parts:

$$L = L_{lm} + \lambda_1 L_{nr} + \lambda_2 L_{ir} \tag{18}$$

where λ_1 and λ_2 are tunable hyper-parameters.

4 Experiment

4.1 Dataset and Evaluation Metrics

We conduct experiments on both ROTOWIRE and RW-FG datasets. They all comprise pairs of NBA basketball game statistics and summaries. There are two main differences between ROTOWIRE and RW-FG. The first is the team statistic table in later containing more numeric records. The other is RW-FG removes the unsupported sentences by the input tables. We use the official training, development, and test splits for both datasets, which are 3,398/727/728 and 5,232/1,125/1,119, respectively.

Model	ROTOWIRE						
	RG		CS			CO	BLEU
	#	P%	P%	R%	F1%	DLD%	
Gold	23.31	94.79	100	100	100	100	100
TEMP	**54.23**	**99.94**	26.99	**58.16**	-	14.92	8.46
CC (Wiseman et al., 2017)	23.72	74.80	29.49	36.18	31.52	15.42	14.19
NCP (Puduppully et al., 2019a)	**34.28**	87.47	34.18	51.22	40.99	18.58	16.50
NCP (Our implementation)	31.95	86.96	33.13	47.59	39.06	17.47	15.26
ENT (Puduppully et al., 2019b)	30.11	92.96	38.67	48.51	43.09	20.17	16.12
HETD (Gong et al., 2019)	31.47	91.46	36.09	48.01	41.21	20.86	16.85
DU (Gong et al., 2020)	29.42	88.05	38.19	49.66	43.18	22.14	16.12
DUV (Gong et al., 2020)	26.94	87.45	40.73	48.78	44.39	23.32	15.92
Ours	32.73	**93.14**	**40.80**	**55.88**	**47.16**	**25.30**	**17.96**
	RW-FG						
Template	**51.80**	**98.89**	23.98	43.96	31.03	10.25	12.09
ENT	35.69	93.72	39.04	49.29	43.57	17.5	21.23
NCP	35.99	94.21	**43.31**	55.15	48.52	23.46	23.86
NCP + TR (Wang, 2019)	37.49	**95.7**	42.90	56.91	48.92	24.47	24.41
Ours	**38.08**	94.75	42.72	**57.56**	**49.04**	**25.23**	**24.52**

Table 1: Automatic evaluation results on the test set. On ROTOWIRE, our results are obtained with Puduppully et al. (2019a)'s updated models. The others are from corresponding papers. On RW-FG, the baselines' results are taken from Wang (2019), and we evaluate directly using the code released by Wang (2019).

Following previous works, we use BLEU and three extractive evaluation metrics, Relation Generation (RG), Content Selection (CS), and Content Ordering (CO) (Wiseman et al., 2017) to evaluate the table-to-text results. More specifically, RG measures the content fidelity of generated text, CS measures how well the generated text matches the reference in selecting which records to generate, and CO measures the ability on context planning. We refer the readers to Wiseman et al. (2017)'s paper for more detailed information on these extractive metrics.

We apply Accuracy (Acc) and normalized Damerau Levenshtein Distance (DLD) (Brill and Moore, 2000) to evaluate the two auxiliary supervision tasks. Accuracy measures the percentage of record sequences for which their absolute positions are correctly predicted (Logeswaran et al., 2018).

4.2 Implementation Details

To make a fair comparison, we follow the configurations in (Puduppully et al., 2019a; Gong et al., 2019). For the table-to-text model, we set word embedding and LSTM decoder hidden size as 600. We set GatedGat's layer as 2 and the numbers of heads as 2. We employ a two-layer LSTM decoder with Input feeding during text generation. We apply dropout at a rate 0.3. For text decoding, we use BPTT and set the truncate size to 100. We set the beam size to 5 during inference. For the two auxiliary tasks, we employ two one-layer LSTM as the decoder and set the LSTM decoder hidden size as 600, respectively. We adjust λ_1 between 0.8 and 1.0, λ_2 between 0.2-0.4. Finally, we set them to 0.9 and 0.25 on ROTOWIRE, 1.0 and 0.4 on RW-FG. For inferring, we use the greedy search algorithm. All experiments are conducted on an NVIDIA Tesla V100. Code of our model can be found at `https://github.com/liang8qi/Data2TextWithAuxiliarySupervision`.

4.3 Baselines

We compare our method with several strong baselines, including:

- TEMP (Wiseman et al., 2017) is a template-based method. We refer the readers to this paper for more detailed information on templates.
- CC (Wiseman et al., 2017) is a standard encoder-decoder system with conditional copy mechanism.
- NCP (Puduppully et al., 2019a) and NCP + TR (Wang, 2019) are two Conditional Copy models with the explicit content planning.

Development				
Model	NR Task		IR Task	
	Acc%	DLD%	Acc%	DLD%
Original	46.43	66.15	7.72	27.29
Separate	89.36	92.63	87.81	91.43
Ours	86.56	90.44	84.07	87.74
Test				
Model	NR Task		IR Task	
	Acc%	DLD%	Acc%	DLD%
Original	46.54	66.02	7.71	26.93
Separate	89.15	92.47	87.60	91.26
Ours	86.54	90.40	83.98	87.68

Table 2: Automatic evaluation of the Number Ranking(NR) task and the Importance Ranking (IR) task on ROTOWIRE development and test datasets.

The latter improves NCP by introducing a table restructure loss.

- ENT (Puduppully et al., 2019b) is a method that creates entity-specific representations and generates text using hierarchical attention over the input table and entity memory.
- HETD (Gong et al., 2019) is a method modeling table from three different dimensions (Row, Column and, Time).
- DU & DUV (Gong et al., 2020): the DU brings the sense of value comparison into content planning. Furthermore, DUV introduces content plan verification into DU.

4.4 Main Results

Automatic Evaluation Our results on the two test datasets are summarized in Table 1. For ROTOWIRE, compared with previous neural models, our method achieves state-of-the-art results on Content Selection (CS), Content Ordering (CO), and BLEU. More specifically, compared with the previous best neural models, we obtain more than 4 improvement on CS-P and achieve the best results on CS-R. This implies our method can generate text that contains more salient records. Compared with **NCP**, **DU**, and **DUV**, our method scores the highest on CO, even without explicitly modeling content selection and planning. This indicates that our model can better organize the records when generating a summary for the input tables. We consider there are two main reasons. The first is that our Reasoning Module can learn a better entity representation on row level. The other is that our proposed two auxiliary tasks can supervise the Record Encoder to learn a number-aware and relative importance-aware record representation. As a result, the data-to-text model can make good content planning by considering the entity's performance and the relative importance of the record.

Model	RG		CS	CO	BLEU
	#	P%	F1%	DLD%	
Our Model	34.37	90.03	44.34	23.64	17.31
- *Series*	32.74	91.56	41.42	21.52	17.19
- *RM*	33.91	89.58	43.71	23.04	16.98
+ *NE*	**38.41**	92.28	44.22	23.16	16.23
+ *NE & IE*	32.85	92.68	45.33	24.49	16.81
+ *NR*	32.47	**93.76**	45.93	24.29	**18.56**
+ *IR*	35.30	92.65	43.34	22.04	17.47
+ *NR & IR*	33.93	92.40	**46.13**	**25.28**	17.68

Table 3: Ablation results for evaluating each component's contribution on ROTOWIRE development set.

As shown in Table 1, the results on RW-FG follow a pattern similar to ROTOWIRE. We notice that all models perform better on RW-FG than on ROTOWIRE. We consider that the improvement comes from the purification of data in RW-FG. Wang (2019) removes the sentences that are not supported by the input tables, which reduces the noise in the text and improves the dataset's quality. Due to this, we can obtain more accurate content planning labels from the dataset to train the models (NCP, NCP+TR) that explicitly model content planning and lead to better performance. Therefore, NCP outperforms ENT on RW-FG. However, the purification may make the task easier because some sentences that do not be supported by the tables directly but can be obtained by reasoning may also be removed. This may weaken the Reasoning Module of our model. Nevertheless, we still outperform the compared baselines.

Table 2 shows our model's performance, which is trained together with the two auxiliary tasks on the two auxiliary tasks. We compare it with two baselines. The first is **Original**, which denotes a method that takes the input record sequence as the outputs. Moreover, we separately train our model on the two auxiliary tasks, denoted as **Separate**. As a result, our model achieves comparable performance to **Separate** and is much better than **Original**, even only using the greedy search at testing. The results indicate that the two auxiliary tasks can help the Record Encoder capture the size relation and relative importance relation among records.

Ablation Study First, we examine the effect of changes in the model structure on the results. From Table 3, **Our Model** means our data-to-text model without two auxiliary tasks. We change the connection mode between the Column-wise Encoder

Model	RG		CS	CO	BLEU
	P%	#	F1%	DLD%	
NCP	86.67	**31.46**	40.02	18.73	15.61
NCP+*HEnc*	87.22	27.36	43.55	22.42	15.83
+ *NR*	89.41	28.54	44.56	23.50	16.17
+ *NR&IR*	**90.96**	27.71	**46.29**	**24.23**	**16.29**

Table 4: Generalization study on ROTOWIRE development set. **HEnc** denotes our hierarchical encoder with Reasoning Module.

	Sup	Contra	Gram	Cohere	Concise
Gold	-11.33	-14.00	14.89	12.88	15.33
NCP	11.33	9.78	-10.44	-8.00	-20.89
ENT	-6.00	**-1.11**	-3.33	-7.11	**8.67**
HETD	0.22	3.56	-5.33	-1.33	-5.11
Ours	5.78	1.78	**4.22**	**3.56**	2.00

Table 5: Human evaluation results.

(**CE**) and the Row-wise Encoder (**RE**) to parallel from series (**- Series**). Moreover, we replace the Reasoning Module with a row-level encoder with the content selection gate (**- RM**), which is proposed by Puduppully et al. (2019a). According to the results, the serial connection and the Reasoning Module contribute to the overall performance because BLEU, CS, and CO drop significantly after subtracting them from the full model.

Furthermore, we investigate the impact of the two auxiliary tasks on table-to-text generation. Table 3 shows that both Number Ranking (NR) and Importance Ranking (IR) tasks can improve our basic model. This indicates that it is necessary to explicitly model the size relation and relative importance relation between records. We notice that the model's performance is degraded on CS-F1 and CO when only the IR task is introduced. On the one hand, we believe this is because the modeling of relative importance relation in the row dimension between records depends heavily on its size relation in the column dimension. On the other hand, the CE cannot accurately capture the size relation between records without direct supervision.

Finally, we compare the method that introduces additional feature vectors of the ranking of number and relative importance to Record Embedding with the two auxiliary tasks. Specifically, we first introduce the embedding of ranking of the number (**+ NE**) and further add the embedding of the relative importance of records (**+ IE**). As shown in the third section in Table 3, the **NE** only improves the model on RG. Moreover when the **IE** is incorporated, the model achieves better performance on almost all metrics. However, the improvement is not as significant as the auxiliary tasks. We believe it may be a better way to effectively capture the accurate semantic representation by introducing auxiliary supervision tasks than adding feature vectors directly.

Generalization Study Our method can be applied to the existing works, especially those that explicitly model content selection and planning (NCP, DUV), to improve their performance. To exam our method's generalization, we combine our method with NCP and conduct experiments on the ROTOWIRE development set. The results are summarized in Table 4. First, we use the released code to retrain the NCP model. And then, we replace the NCP's content selection encoder with our hierarchical encoder. As can be seen, our hierarchical encoder with the Reasoning Module improves the NCP model on almost all evaluation metrics. Moreover, we train the model with the proposed two auxiliary supervision tasks. The performance of the model is further improved. This indicates that our method has a good generalization, as it can be easily adapted to other methods and improve their performance.

Human Evaluation To examine whether human judgments corroborate improvements in automatic evaluation metrics, we conducted a human evaluation. Three graduate students with basketball background knowledge and good English reading ability were invited to conduct the evaluation. We compared our best performing model against Gold, NCP, ENT, and HETD. Specifically, we randomly selected 30 games from the test set, and each game is rated by three workers. For each game, we arranged every 5-tuple of summaries into ten pairs. Given each pair, the participants were asked to choose which one is better according to five criteria: Supporting (does the summary contain more supported facts?), Contradicting (does the summary contain more contradicting facts?), Grammaticality (is the summary fluent and grammatical?), Coherence (do the sentences, in summary, follow a coherent discourse?), and Conciseness (does the summary contain less redundant information and repetitions?). Following previous work (Puduppully et al., 2019a), we calculated a model's score for each criterion as the difference between the per-

centage of times when the model is chosen as the best and the percentage of times when the model is chosen as the worst.

The results are summarized in Table 5. As can be seen, the gold texts have significant advantages in contradicting, grammaticality, coherence, and conciseness. Compared with other neural methods, our method receives the highest scores in coherence and grammaticality. This implies that our method can generate texts that contain well-organized facts. Though the ENT model outperforms our model in contradicting and conciseness, our method can be easily applied to it, which we leave for future work.

5 Conclusion

In this work, we mainly make two contributions. The first one is we introduce a reasoning module into a hierarchical table encoder, which enables the model reasoning ability. Moreover, we present to utilize the different auxiliary supervision tasks to help the encoder capture the different relations between records. In detail, the Number Ranking (NR) task is proposed to supervise the Column-wise Encoder to model the numeric size relation between records in the same column. And the Importance Ranking (IR) task helps the Row-wise Encoder capture the relative importance between records in the same row. Experimental results conducted on ROTOWIRE and RW-FG datasets demonstrate the effectiveness of our method. Furthermore, we migrate our method to the NCP model and significantly improve its performance on ROWTOWIRE. This indicates that our proposed method has a good generalization.

References

Yang Bai, Ziran Li, Ning Ding, Ying Shen, and Hai-Tao Zheng. 2020. Infobox-to-text generation with tree-like planning based attention network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3773–3779. ijcai.org.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5020–5027. AAAI Press.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong. Association for Computational Linguistics.

Shuang Chen, Jinpeng Wang, Xiaocheng Feng, Feng Jiang, Bing Qin, and Chin-Yew Lin. 2019. Enhancing neural data-to-text generation models with external background knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3022–3032, Hong Kong, China. Association for Computational Linguistics.

Heng Gong, Wei Bi, Xiaocheng Feng, Bing Qin, Xiaojiang Liu, and Ting Liu. 2020. Enhancing content planning for table-to-text generation with data understanding and verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2905–2914, Online. Association for Computational Linguistics.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.

Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Preksha Nema, Mitesh M. Khapra, and Shreyas Shetty. 2018. A mixed hierarchical attention based encoder-decoder approach for standard table summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 622–627, New Orleans, Louisiana. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA,*

January 27 - February 1, 2019, pages 6786–6793. AAAI Press.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4881–4888. AAAI Press.

Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5285–5292. AAAI Press.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Preksha Nema, Shreyas Shetty, Parag Jain, Anirban Laha, Karthik Sankaranarayanan, and Mitesh M. Khapra. 2018. Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1539–1550, New Orleans, Louisiana. Association for Computational Linguistics.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. 2018. Operation-guided neural networks for high fidelity data-to-text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3879–3889, Brussels, Belgium. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with content selection and planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6908–6915. AAAI Press.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5414–5421. AAAI Press.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2020. Sentence generation for entity description with content-plan attention. In *AAAI*, pages 9057–9064.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Hongmin Wang. 2019. Revisiting challenges in data-to-text generation with fact grounding. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 311–322, Tokyo, Japan. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

A Development Performance

We present the performance of the compared baselines and our model on ROTOWIRE[1] and RW-FG[2] development sets in Table 6. As can be seen, the test datasets' results in Table 1 follow a pattern similar to the development sets.

[1]https://github.com/harvardnlp/boxscore-data

[2]https://github.com/wanghm92/rw_fg

Model	ROTOWIRE						
	RG		CS			CO	BLEU
	#	P%	P%	R%	F1%	DLD%	
Gold	23.34	94.79	100	100	100	100	100
TEMP	**54.29**	**99.92**	26.61	**59.16**	36.69	14.42	8.51
CC	23.95	75.10	28.11	35.86	31.52	15.33	14.57
NCP	33.88	87.51	33.52	51.21	40.52	18.57	16.19
ENT	30.39	91.98	36.62	48.18	41.62	19.66	15.97
HETD	32.11	91.84	35.39	48.98	41.09	20.70	16.24
DU	28.81	87.23	39.03	51.64	44.46	22.97	16.64
DUV	26.11	87.35	**42.00**	50.63	45.91	24.86	16.29
Ours	**33.93**	**92.40**	38.65	**57.2**	**46.13**	**25.28**	**17.68**
	RW-FG						
Template	**51.81**	**99.09**	23.78	43.75	30.81	10.06	11.96
ENT	35.56	93.3	39.04	40.19	50.17	17.81	21.67
NCP	36.28	94.27	**43.31**	55.96	48.91	24.08	24.49
NCP + TR	37.04	**95.65**	43.09	57.24	49.17	24.75	**24.80**
Ours	**38.50**	94.35	42.88	**58.16**	**49.52**	**25.30**	24.62

Table 6: Automatic evaluation results on development sets.

Model	RG		CS			CO	BLEU
	#	P%	P%	R%	F1%	DLD%	
HEnc	34.37	90.03	36.75	55.87	44.34	23.64	17.31
+ *A-NR*	35.20	92.03	37.51	57.3	45.34	24.17	17.64
+ *A-NR & D-IR*	**36.73**	90.96	37.46	**58.67**	45.72	24.77	17.27
+ *A-NR & A-IR*	35.00	92.38	38.15	57.17	45.76	24.82	17.38
+ *D-NR*	33.38	93.76	39.05	55.74	45.93	24.29	**18.56**
+ *D-NR & A-IR*	33.93	**92.40**	38.65	57.2	**46.13**	**25.28**	17.68
+ *D-NR & D-IR*	32.47	91.05	**39.22**	55.83	46.08	24.97	18.01

Table 7: Impact of different settings of Number Ranking (NO) and Importance Ranking (SO). HEnc denotes our data-to-text model, which incorporates a Reasoning Module. The prefixes A and D denote ascending and descending operations, respectively.

B Impact of different Ranking Directions

We also explore the impact of different settings for Number Ranking and Importance Ranking on the data-to-text model. The results are summarized in Table 7. We observe that compared with the basic model, almost all the settings can improve the data-to-text model on Content Selection(CS), Content Ordering(CO), and BLEU. This indicates the proposed two tasks are effective and robust.