Enhancing Content Planning for Table-to-Text Generation with Data Understanding and Verification

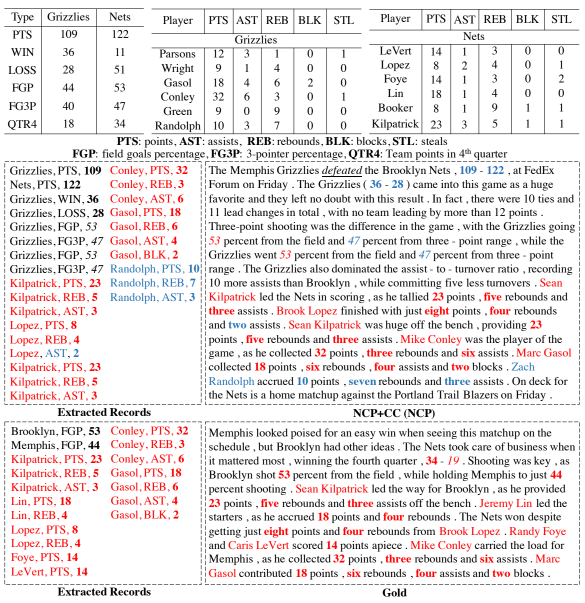
Heng Gong,[1] **Wei Bi,** **Xiaocheng Feng,**[1] **Bing Qin,**[1] **Xiaojiang Liu,** **Ting Liu**[1]
[1]Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
{hgong,xcfeng,qinb,tliu}@ir.hit.edu.cn
weibi@cse.ust.hk xiaojiangliu84@hotmail.com

Abstract

Neural table-to-text models, which select and order salient data, as well as verbalizing them fluently via surface realization, have achieved promising progress. Based on results from previous work, the performance bottleneck of current models lies in the stage of content planning (selecting and ordering salient content from the input). That is, performance drops drastically when an oracle content plan is replaced by a model-inferred one during surface realization. In this paper, we propose to enhance neural content planning by (1) understanding data values with contextual numerical value representations that bring the sense of value comparison into content planning; (2) verifying the importance and ordering of the selected sequence of records with policy gradient. We evaluated our model on ROTOWIRE and MLB, two datasets on this task, and results show that our model outperforms existing systems with respect to content planning metrics.

1 Introduction

Table-to-text generation refers to the task of generating text from structured data. Models for this task can be mainly categorized into two types: pipeline-style models, which decompose the generation process into sequential stages, including content planning (Stage 1, selecting and ordering salient content from the input) and surface realization (Stage 2, converting the content plan to surface string) (Kukich, 1983; McKeown, 1985); and end-to-end models, which entangle aforementioned stages and generate text directly from structured data through a neural encoder-decoder framework (Wiseman et al., 2017; Nie et al., 2018). As in Fig. 1, this task provides tables with redundant records. Each record has three elements: table row header (entity, e.g. Conley), table column header (type, e.g. points) and table cell (value, e.g. 32). Models are expected to generate descriptive text



Type	Grizzlies	Nets
PTS	109	122
WIN	36	11
LOSS	28	51
FGP	44	53
FG3P	40	47
QTR4	18	34

Player	PTS	AST	REB	BLK	STL
Grizzlies					
Parsons	12	3	1	0	1
Wright	9	1	4	0	0
Gasol	18	4	6	2	0
Conley	32	6	3	0	1
Green	9	0	9	0	0
Randolph	10	3	7	0	0

Player	PTS	AST	REB	BLK	STL
Nets					
LeVert	14	1	3	0	0
Lopez	8	2	4	0	1
Foye	14	1	3	0	2
Lin	18	1	4	0	0
Booker	8	1	9	1	1
Kilpatrick	23	3	5	1	1

PTS: points, **AST**: assists, **REB**: rebounds, **BLK**: blocks, **STL**: steals
FGP: field goals percentage, **FG3P**: 3-pointer percentage, **QTR4**: Team points in 4th quarter

Grizzlies, PTS, **109**	Conley, PTS, **32**
Nets, PTS, **122**	Conley, REB, **3**
Grizzlies, WIN, **36**	Conley, AST, **6**
Grizzlies, LOSS, **28**	Gasol, PTS, **18**
Grizzlies, FGP, *53*	Gasol, REB, **6**
Grizzlies, FG3P, *47*	Gasol, AST, **4**
Grizzlies, FGP, *53*	Gasol, BLK, **2**
Grizzlies, FG3P, *47*	Randolph, PTS, **10**
Kilpatrick, PTS, **23**	Randolph, REB, **7**
Kilpatrick, REB, **5**	Randolph, AST, **3**
Kilpatrick, AST, **3**	
Lopez, PTS, **8**	
Lopez, REB, **4**	
Lopez, AST, **2**	
Kilpatrick, PTS, **23**	
Kilpatrick, REB, **5**	
Kilpatrick, AST, **3**	

Extracted Records

The Memphis Grizzlies *defeated* the Brooklyn Nets , **109** - **122** , at FedEx Forum on Friday . The Grizzlies (**36** - **28**) came into this game as a huge favorite and they left no doubt with this result . In fact , there were 10 ties and 11 lead changes in total , with no team leading by more than 12 points . Three-point shooting was the difference in the game , with the Grizzlies going *53* percent from the field and *47* percent from three - point range , while the Grizzlies went *53* percent from the field and *47* percent from three - point range . The Grizzlies also dominated the assist - to - turnover ratio , recording 10 more assists than Brooklyn , while committing five less turnovers . Sean Kilpatrick led the Nets in scoring , as he tallied **23** points , **five** rebounds and **three** assists . Brook Lopez finished with just **eight** points , **four** rebounds and **two** assists . Sean Kilpatrick was huge off the bench , providing **23** points , **five** rebounds and **three** assists . Mike Conley was the player of the game , as he collected **32** points , **three** rebounds and **six** assists . Marc Gasol collected **18** points , **six** rebounds , **four** assists and **two** blocks . Zach Randolph accrued **10** points , **seven** rebounds and **three** assists . On deck for the Nets is a home matchup against the Portland Trail Blazers on Friday .

NCP+CC (NCP)

Brooklyn, FGP, **53**	Conley, PTS, **32**
Memphis, FGP, **44**	Conley, REB, **3**
Kilpatrick, PTS, **23**	Conley, AST, **6**
Kilpatrick, REB, **5**	Gasol, PTS, **18**
Kilpatrick, AST, **3**	Gasol, REB, **6**
Lin, PTS, **18**	Gasol, AST, **4**
Lin, REB, **4**	Gasol, BLK, **2**
Lopez, PTS, **8**	
Lopez, REB, **4**	
Foye, PTS, **14**	
LeVert, PTS, **14**	

Extracted Records

Memphis looked poised for an easy win when seeing this matchup on the schedule , but Brooklyn had other ideas . The Nets took care of business when it mattered most , winning the fourth quarter , **34** - *19* . Shooting was key , as Brooklyn shot **53** percent from the field , while holding Memphis to just **44** percent shooting . Sean Kilpatrick led the way for Brooklyn , as he provided **23** points , **five** rebounds and **three** assists off the bench . Jeremy Lin led the starters , as he accrued **18** points and **four** rebounds . The Nets won despite getting just **eight** points and **four** rebounds from Brook Lopez . Randy Foye and Caris LeVert scored **14** points apiece . Mike Conley carried the load for Memphis , as he collected **32** points , **three** rebounds and **six** assists . Marc Gasol contributed **18** points , **six** rebounds , **four** assists and **two** blocks .

Gold

Figure 1: A ROTOWIRE's example with NCP's result and gold text. Important/unimportant entities and records are in red/blue. Text that accurately/incorrectly report statistics in table is in bold/italic.

reflecting salient records. Many neural end-to-end models have achieved remarkable progress of generating fluent and natural text on this task (Puduppully et al., 2019b; Gong et al., 2019).

However, previous work notices that the content planning stage is the key factor in table-to-text generation (Gkatzia, 2016), but end-to-end models are difficult to explicitly improve their content planning ability. Recently, Puduppully et al. (2019a) proposed Neural Content Planning (NCP), a two-stage model that explicitly selects and orders salient records whilst keeping the ability to generate fluent text of end-to-end models. They show that content planning (referring to both "content selection and planning" in Puduppully et al. (2019a)) indeed correlates with the quality of final output. Yet, NCP simply maximizes the log-likelihood of pre-extracted sequences of content plans given all records. According to their reported results, the inferred content plans are still far from the oracle. Thus, we focus on bridging the gap between the inferred content plans and the upper-bounds in Stage 1, and thus improving the final generation results.

We observe that whether a record is important highly depends on its record value. However, NCP, as well as other neural generation models, treats numerical values in table as tokens, and the prominent role of values in content planning is not recognized. Let's take Fig. 1 for example. Compared to the gold text, NCP mistakenly states that "The Memphis Grizzlies defeated the Brooklyn Nets" while "Nets" clearly score more points than "Grizzlies" in this match. Also, NCP neglects important players such as "Lin" who performs the second best in team "Nets". We hypothesize that this is because the model lacks understanding of values in their given context (here context means the structured table information) when representing corresponding records. In addition, we find that NCP tends to include redundant information when describing those players. For example, NCP includes redundant "two assists" when describing "Lopez". A possible reason is that the use of maximum likelihood estimation (MLE) is not enough to help verify important records during training.

To address the aforementioned numeric value understanding and important record verification problems, we propose a generation model with Data Understanding and Verfication (DUV), improving content planning in the framework of NCP. Specifically, we design contextual numeric value representations obtained through a pre-trained ranking task. In the pre-trained model, we compare pairwise numerical values describing the same type of information and decide which has a higher value. In the record encoder when training the model, we replace the value representation with its contextual version from the pre-trained model. In this way, the constructed record representation is also context-aware. Besides, instead of using the simple MLE, we design integrated rewards to verify content planning results. We conducted experiments on ROTOWIRE and MLB, showing that our model outperforms existing systems regarding the *content selection* and *ordering* metric.

2 Background

This task's input consists of tables S of records. The basics of a record r include entity $r.e$, type $r.c$, value $r.v$ and features $r.f$. Models need to generate text $y = (y_1, y_2, ..., y_{|y|})$ ($|y|$ is number of words) to describe important records in tables. As stated in Sec.1, this task has two main stages: (i) content planning, and (ii) surface realization. Puduppully et al. (2019a) propose Neural Content Planning (NCP) to explicitly optimize these two stages in deep neural networks, making the generation process more interpretable with an intermediate content plan. Thus, we use it as base model.

In Stage 1 (content planning), NCP embeds tokens into embedding vectors and encodes each record r with one-layer MLP for ROTOWIRE:

$$\mathbf{r} = \text{ReLU}(\mathbf{W}_a[\mathbf{r}.e; \mathbf{r}.c; \mathbf{r}.v; \mathbf{r}.f] + \mathbf{b}_a). \quad (1)$$

Here, $\mathbf{r}.*$ represents their embedding vectors. $\mathbf{W}_a$ and $\mathbf{b}_a$ are trainable parameters and $[;]$ denotes vector concatenation. The reason to choose MLP is that its records are game statistics without sequential relationship between records. For MLB, we follow Puduppully et al. (2019b) and use LSTM instead because its input includes sequential event data. Next, a content selection gate is applied on each $\mathbf{r}$ to control the amount of information flowing from the record r. A LSTM-based pointer network (Vinyals et al., 2015) is applied to sequentially decode a content plan, which is a sequence of important records extracted from the output text, denoted as $r^* = \{r_1^*, \dots, r_T^*\}$ (T is the number of records mentioned in y). Here, we follow Puduppully et al. (2019a) to extract content plans using an information extraction (IE) approach as oracles. In each time step, the decoder takes previously selected record's representation as input and use the attention weights to select the next important one.

In Stage 2 (surface realization), a standard encoder-decoder model is applied, taking the output content plan from Stage 1 as input and generating text with attention mechanism (Luong et al., 2015) and conditional copy mechanism (Gulcehre et al., 2016). From results in Puduppully et al. (2019a), it is observed that performance bottleneck lies in Stage 1. That is, if we feed gold content plans into Stage 2, final results are much better, but if inferred content plans are fed instead, performance decreases drastically. Therefore, we focus on improving NCP's Stage 1 for better final outputs.

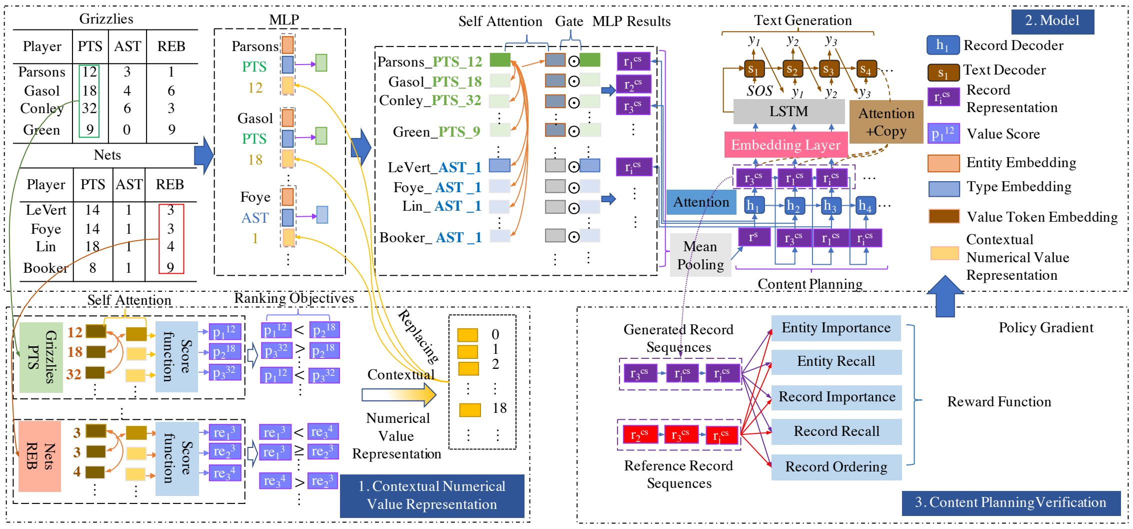


Figure 2: The architecture of our approach. To enhance neural content planning in the base model (Module 2), we propose two modules: Contextual Numerical Value Representation (Module 1) and Content Planning Verification (Module 3). First, we use Module 1 to pre-train contextual numerical value representation and replace the numerical value's token embedding. After Module 2 planning records from table, we use Module 3 to optimize with reinforced supervision signals. Then, Module 2 use the content plans to generate text.

3 Approach

We propose to improve content planning (Stage 1 of NCP) from two aspects: (i) during record encoding, we design a contextual numeric value representation to improve the understanding of entities' (players' and teams') performance; (ii) a reinforced training strategy with targeted supervision signals is used to compensate maximizing the MLE in pointer network to boost model's content planning ability. Fig. 2 illustrates the overall training procedure. We first pre-train a model to learn contextual numeric value representations to understand relationship between records' numeric values by pairwise ranking loss. Secondly, given the pre-trained model and table S, we encode each record with its contextual numeric value representation. In decoding phase of Stage 1, the pointer network is guided to favor important records for content planning with the help of reinforced supervision signals. Stage 2 remains the same as in the base model. We describe details in following parts.

3.1 Contextual Numerical Value Representation

Current table-to-text models treat numerical values in table as tokens and use embeddings to represent them. However, a numerical value has some attributes that a text token doesn't have. Generally, a larger numerical value indicates better performance of a player. Also, considering different context, an identical numerical value can convey different meaning. (1) One numerical value describing same type of records can correspond to different situations on court. For instance, if a player got "23" points in a game, top 1 among all players, it indicates outstanding performance. But, if there are other players on court with points over 30, it becomes less outstanding. (2) The same numerical value describing different types of information should not be interpreted in the same way. For example, "5" assists may indicate good performance, while "5" points may suggest disappointing performance. Hence, it is important to model a numerical value in context of other numerical values describing the same type of information in order to understand what is behind those numerical values. Here, we propose to learn contextual numerical value representations for this task.

We extract numerical values that describe the same type of information from the same table to form training samples (e.g. players' points in Nets) for a pre-trained task. Our main idea is to use transformer encoder (Vaswani et al., 2017) to compare each numerical value with others in each training sample. We first use it to fuse information of numerical values in the same sample and obtain their contextual numerical value representations. Next, we optimize the pairwise ranking loss using their contextual representations such that a large numerical value is with a higher ranking score. Taking all raw numerical value embedding $\mathbf{r}_i.v$'s of each training sample as input, we construct the contextual numerical value embeddings $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_n]$ via multi-layer transformer encoder:

$$
\begin{aligned}
\mathbf{H}^0 &= [\mathbf{r}_1.v, \ldots, \mathbf{r}_i.v, \ldots, \mathbf{r}_n.v], && (2)\\
\mathbf{A}^k &= \text{LN}(\mathbf{H}^{k-1} + \text{MHSelfAtt}(\mathbf{H}^{k-1})) && (3)\\
\mathbf{H}^k &= \text{LN}(\mathbf{A}^k + \text{FFN}(\mathbf{A}^k)),\ \tilde{\mathbf{R}} = \mathbf{H}^K && (4)
\end{aligned}
$$

where n is the number of numerical values in the sample, LN is the layer normalization, MHSelfAtt is the multi-head self-attention function, and FFN is position-wise feed-forward network.

Given a pair of contextual numerical value representations $\tilde{\mathbf{r}}_i$ and $\tilde{\mathbf{r}}_j$, we use a fully connected layer $f(\tilde{\mathbf{r}}_i) = \text{sigmoid}(\mathbf{W}_p\tilde{\mathbf{r}}_i + \mathbf{b}_p)$ to calculate the ranking score for each numerical value in the current input sample. If $r_i.v \geq r_j.v$, we expect $f(\tilde{\mathbf{r}}_i)$ to be higher than $f(\tilde{\mathbf{r}}_j)$. For training contextual numerical value representations, we use the hinge loss (Eq.5). ξ is the margin and $T(\cdot)$ gives $+1$ if $\cdot$ is true and -1 otherwise.

$$
\begin{aligned}
\ell_{pre} &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} \max(0, \xi - T(r_i.v \geq r_j.v) \\
&\quad (f(\tilde{\mathbf{r}}_i) - f(\tilde{\mathbf{r}}_j))) && (5)
\end{aligned}
$$

We construct training samples of the pre-trained task using all training tables. Note that numerical values from different types of information form different samples. When the pre-trained model is converged, we use it in the record encoder in Eq. 1 by replacing the token embedding $\mathbf{r}_i.v$ with its contextual representation $\tilde{\mathbf{r}}_i$ via Eq. 2 to Eq. 4.

3.2 Content Planning Verification

The original NCP uses the pointer network to explicitly infer a content plan by optimizing the MLE of gold content plans. As noticed in other generation tasks (Sordoni et al., 2015; Li et al., 2016a; Dai et al., 2017), generation models with the MLE as the objective function tend to generate universal output sequences observed in the training data and it is desirable to integrate developer-defined rewards that better mimic the true goal of an ideal output sequence (Li et al., 2016b), which is the sequence of the content plan in our task. In order to explicitly reflect the quality of content plans, we explore rewards that measure the following five criteria, and optimized the model according to them via policy gradient (Sutton and Barto, 1998).

- Entity Importance (EI) evaluates if a predicted record r_t contains an important entity by comparing whether the entity is mentioned in the gold content plan $\{r_i^*\}$. $R(\cdot)$ function gives +1 reward when $\cdot$ is true and -1 otherwise.

$$\text{EI}(r_t) = R(r_t.e \in \{r_i^*.e\}). \qquad (6)$$

- Entity Recall (ER) measures how many important entities are covered by the decoded content plan $r = \{r_t\}$. $\mathbf{1}(\cdot)$ is the indicator function which is 1 when $\cdot$ is true, otherwise 0.

$$\text{ER}(r) = \frac{1}{|\{r_i^*.e\}|}\sum_{i=1}^{|\{r_i^*.e\}|} \mathbf{1}[r_i^*.e \in \{r_t.e\}]. \qquad (7)$$

- Record Importance (RI) and Record Recall (RR) are similar to EI and ER respectively but focus on each individual record instead of entity only:

$$
\begin{aligned}
\text{RI}(r_t) &= R(r_t \in \{r_i^*\}) && (8)\\
\text{RR}(r) &= \frac{1}{|\{r_i^*\}|}\sum_{i=1}^{|\{r_i^*\}|} \mathbf{1}[r_i^* \in \{r_t\}]. && (9)
\end{aligned}
$$

- Record Ordering (RO) calculates the normalized Damerau-Levenshtein Distance (Brill and Moore, 2000) between the predicted content plan r and the reference r^* in order to measure how well the model organizes the chosen records.

The above designed rewards measure the content plan on different granularity. EI and ER focus on whether the selected entity (player/team) is an important one. It is also crucial to decide which of the entity's records are needed to be mentioned. Therefore, we also include RI and RR. Afterwards, we sample record sequence, combine all rewards and use policy gradient to guide the optimization of content selection given S as the input table:

$$
\begin{aligned}
\mathcal{L}_{rl} &= -\frac{1}{T}\sum_{t=1}^{T} R_{tok}\log P(r_t|r_{<t}, S) \\
&\quad -\frac{1}{T}(R_{seq}-\beta)\log P(r|S) && (10)\\
R_{tok} &= \gamma_1\text{EI}(r_t) + \gamma_2\text{RI}(r_t) && (11)\\
R_{seq} &= \gamma_3\text{ER}(r)+\gamma_4\text{RR}(r)+\gamma_5\text{RO}(r) && (12)
\end{aligned}
$$

Given a batch of input tables $\{S\}_G$ and gold content plan $\{r^*\}_G$, we first train the pointer network by optimizing the MLE: $\mathcal{L}_{gen} = -\frac{1}{G}\sum_{g=1}^{G}\frac{1}{T_g}\sum_{t=1}^{T_g}\log P(r_{t,g}^*|r_{<t,g}^*, S_g)$. Then, we further finetune it with both the MLE loss and policy gradient: $\mathcal{L} = \gamma_6\mathcal{L}_{rl} + (1-\gamma_6)\mathcal{L}_{gen}$. Please note that T represents length of the content plan. γ_1-γ_6 and β are hyper-parameters.

4 Experiments

4.1 Setup

Dataset and Evaluation Metrics We conducted experiments on both ROTOWIRE[1] and MLB (Puduppully et al., 2019b) dataset. The former provides pairs of NBA game statistics and summary. Differently, the latter provides summary and **heterogeneous** input, consisting of MLB game statistics and event data (including event type, actors, etc.) in chronological order. For ROTOWIRE, we follow official training, development and test splits of 3398/727/728 instances. For MLB, as the contents are not released, we are able to retrieve a split of 22820/1739/1744 instances via official scripts [2].

For evaluations, we use BLEU (Papineni et al., 2002) and three extractive metrics, which evaluate the generated results from the following aspects: (1) Relation Generation (RG), measuring the text fidelity about whether to describe information from table truthfully. (2) Content Selection (CS) to measure whether important information is selected from redundant game statistics. (3) Content Ordering (CO) evaluates a model's ability to plan and order data records naturally in text. More details can be found in Wiseman et al. (2017).

Implementation Details We follow Puduppully et al. (2019a)'s and Puduppully et al. (2019b)'s training configurations in the base model for ROTOWIRE and MLB respectively. We chose the proposed hyper-parameters based on performance on development set. Due to page limit, we include model and training details in Appendix. Codes of our model can be found at https://github.com/ErnestGong/data2text-duv.

4.2 Results

Comparing Methods In this section, we compare:

• Template: We follow Wiseman et al. (2017) and Puduppully et al. (2019b) for constructing template-based generators for ROTOWIRE and MLB respectively. The details and Conditional Copy (CC) model can be found in those papers.

• NCP+CC (NCP): our base model. Here, we provide both results reported in the original paper and reproduced by us, denoted as NCP(R). We also try a variant of NCP by using separate sets of embeddings in the encoders of two stages, denoted as S-NCP. We observe that S-NCP is comparable with reproduced NCP, with the ability to explicitly improve Stage 1 without affecting Stage 2. Thus. we use it to further verify our proposed model.

• Entity Modeling (ENT) (Puduppully et al., 2019b) and Hierarchical Encoder on Three Dimensions (HETD) (Gong et al., 2019) are two state-of-the-art models on ROTOWIRE and/or MLB. OpAtt (Nie et al., 2018) introduces pre-executed operations for text generation.

• Data Understanding with content plan Verification (DUV): our proposed full model. We also include two variants for ablations: S-NCP + Verification (S-N+V) to study our model without data understanding, and Data Understanding (DU) to study without content plan verification.

Automatic Evaluation For ROTOWIRE, as shown in Table 1, template system achieves high RG P% (high-fidelity) due to rigid rules. Also, it achieves high CS R% since it includes vast amount of information (high RG #) and some of which are redundant (low CS P%). Compared with it, most neural models perform significantly better at filtering redundant records (CS P%) while still covering many important records, leading to better CS F1%. The higher CO also shows that neural models can better organize data records conditioned on the data. Among all neural models, DUV exceeds other neural models in terms of content selection (CS F1%) and content ordering (CO) on test set. Also, by comparing DUV with its base model (S-NCP), our model improves more on CS P%. In terms of RG, our model also performs better than base model, but still has a gap to ENT and HETD. This is mainly affected by surface realization (Stage 2), which is beyond the scope of this paper.

For MLB, we find similar pattern as discussed above. The differences are (1) improvements on CS and CO are less significant than on ROTOWIRE. Since MLB includes additional event data that ROTOWIRE doesn't have, we separate out the statistical data in Table 4 for fair comparison. We find that base model (S-NCP) achieves 73.43% (Table 4) regarding statistical data on MLB v.s. 44.37% (Table 3) on ROTOWIRE of CS F1% in Stage 1, leaving much less room for improvement. (2) NCP-style models achieve less BLEU than ENT on MLB. The latter (Brevity Penalty, BP 0.736) generates longer text compared with DUV (BP 0.623). This is mainly due to surface realization (Stage 2), which we leave for future work.

Table 1 also includes ablations of our model

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

[2]https://github.com/ratishsp/mlb-data-scripts

ROTOWIRE (RW)	RG		CS			CO	BLEU
	P%	**#**	**P%**	**R%**	**F1%**	**DLD%**	
TEMP	**99.94**	**54.21**	27.02	**58.22**	36.91	15.07	8.58
ED+CC	74.80	23.72	29.49	36.18	32.49	15.42	14.19
OpAtt	-	-	-	-	-	-	14.74
NCP+CC (NCP)	87.47	34.28	34.18	51.22	41.00	18.58	16.50
ENT	92.69	30.11	38.64	48.51	43.02	20.17	16.12
HETD	91.46	31.47	36.09	48.01	41.21	20.86	**16.85**
NCP(R)	86.06	26.60	36.56	43.57	39.76	18.84	14.84
S-NCP	85.05	26.93	35.59	43.76	39.25	18.51	14.63
S-NCP+V (S-N+V)	85.29	25.36	37.12	42.82	39.77	18.99	13.77
DU	88.05	29.42	38.19	49.66	43.18	22.14	16.12
DUV	87.45	26.94	**40.73**	48.78	**44.39**	**23.32**	15.92

MLB	RG		CS			CO	BLEU
	P%	**#**	**P%**	**R%**	**F1%**	**DLD%**	
TEMP	**97.99**	**57.11**	23.51	**65.69**	34.63	10.80	2.80
ED+CC	91.74	17.10	63.45	47.27	54.18	25.59	9.65
NCP+CC (NCP)	88.65	15.96	**64.16**	51.47	57.12	27.11	8.39
ENT	84.61	22.10	55.32	60.92	57.99	23.59	**13.11**
S-NCP	87.80	16.67	62.63	53.56	57.74	27.22	9.62
S-NCP+V (S-N+V)	88.13	16.73	62.89	53.91	58.06	27.69	9.54
DU	87.99	16.63	62.80	53.75	57.93	27.47	9.53
DUV	89.02	16.65	63.44	53.63	**58.12**	**27.78**	9.51

Table 1: Automatic evaluation results on test set. On ROTOWIRE (*top*), results are obtained with updated extractive evaluation models (Puduppully et al., 2019a). Those above the dash line, except for TEMP, are from corresponding papers. On MLB (*bottom*), since our vocabulary is different from the one in released models, we re-train the Information Extraction (IE) model via official script (Puduppully et al., 2019b) on re-collected dataset. It can recall 96.60% of tuples with precision of 96.39% on test set, compared to the released oracle tuples. All baselines' results on MLB are reproduced by us. Note that for ENT, we directly use the released code to train.

(S-N+V and DU). Results show that both data understanding and verification modules contribute to the overall improvement. Due to page limit, we include validation performance in Appendix.

Human Evaluation Each example below is evaluated by 3 different annotators from a commercial annotation company, who are proficient in English and we report the average of three annotators' results in following settings. First, We sample 30 examples from test set and asked annotators to determine how many information in the summary are correct (#Sup) and how many are contradicting (#Cont) to the table. On ROTOWIRE, our model describes the table more concisely (closest #Sup to gold text) while produces significantly less contradicting facts than NCP thanks to significant improvement on Stage 1. We observe that gold text contains incorrect facts (e.g. wrong field-goal percentage) while #Cont of TEMP is due to annotation error. Gap between ENT and DUV on #Cont shows potential of Stage 2, which is beyond the scope of this paper.

Second, we arrange results from models of each example into 10/15 pairs (ROTOWIRE/MLB) and asked annotators to determine which one in the pair performs better in terms of grammaticality, coherence and conciseness. The reported result is the subtraction of the percentage of time a system is considered better and when considered worse. On ROTOWIRE, DUV can generate most coherent text among neural models, but less satisfying on grammaticality and conciseness, compared with ENT. This is mainly affected by surface realization (Stage 2). A possible way is to use large-scale pre-trained language models such as GPT-2 (Radford et al., 2019) to address this issue. In MLB, DUV achieves comparable performance with NCP across 5 metrics due to the same Stage 2.

RW	#Sup	#Cont	Gram	Coher	Conc
Gold	31.91	1.92⋆	26.44⋆	6.44	8.89⋆
TEMP	53.87⋆	0.02⋆	-10.67	2.67	29.11⋆
NCP	40.61⋆	6.38⋆	-20.44⋆	-8.22⋆	-24.00⋆
ENT	35.06	2.69	8.22⋆	-2.67	-5.78
DUV	30.74	3.61	-3.56	1.78	-8.22

MLB	#Sup	#Cont	Gram	Coher	Conc
Gold	15.27	4.30	26.67⋆	28.74⋆	32.59⋆
TEMP	52.02⋆	0.72⋆	-15.56⋆	-19.11⋆	14.81⋆
CC	14.04	1.99⋆	-9.48	-9.33	-22.52⋆
NCP	13.82	2.87	-1.04	-3.11	-9.33
ENT	18.89⋆	3.29	4.89⋆	7.85⋆	-7.70
DUV	13.24	3.80	-5.48	-5.04	-7.85

Table 2: Human evaluation results. Models with ⋆ perform significantly different from DUV ($p < 0.05$), using a one-way ANOVA with posthoc Tukey HSD tests. We omit CC on ROTOWIRE because NCP is proven to be better (Puduppully et al., 2019a).

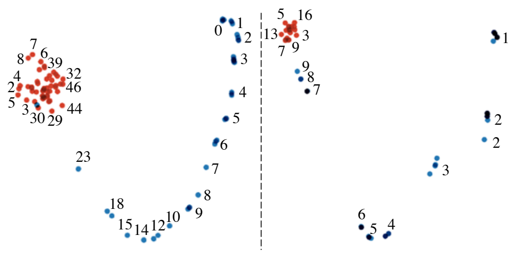


Figure 3: Visualization of contextual numerical value representations of PTS, AST, REB of the example in Fig. 1 obtained by our pre-trained model (in blue) and the token embeddings of values from 0 to 49 (in red) via PCA (Pearson, 1901) on *left* (ROTOWIRE). The *right* one are obtained on a MLB's example (Appendix).

4.3 Analysis

Visualization Fig. 3 visualizes value's token embeddings (in red) and our contextual numerical value representations (in blue). Token embeddings are closer between each other while the contextual ones are more discriminative and naturally ordered from low to high along the "blue arc". We hypothesize this phenomenon contributes to the improvement of content selection.

Content Planning In Table 3, we discuss Stage 1's content planning (CS and CO) results on ROTOWIRE. DU improves on all metrics. It focuses more on covering important records (CS R%) compared with others. By adding verification on top of DU (DUV), it can further improve on CS P%, F1%

RW	CS P%	CS R%	CS F1%	CO
NCP	38.00	53.72	44.51	20.27
NCP(R)	41.43	48.05	44.50	21.49
S-NCP	40.28	49.39	44.37	21.28
S-N+V	42.52	48.13	45.15	21.34
DU	43.38	**54.48**	48.30	24.42
DUV	46.97	53.93	**50.21**	**26.63**
-EI	46.75	53.69	49.98	26.05
-ER	46.84	53.36	49.89	26.26
-RI	46.70	54.02	50.09	26.18
-RR	47.00	53.85	50.19	26.41
-RO	**47.01**	53.67	50.12	26.26

Table 3: Results of Stage 1 performance on content planning metrics on ROTOWIRE's development set.

and CO. Considering both CS P% and R%, DUV can generate more concise but informative content plans with little sacrifice on recall.

Next, by subtracting each reward from DUV, we observe that all rewards contribute to DUV's improvement on content selection and ordering.

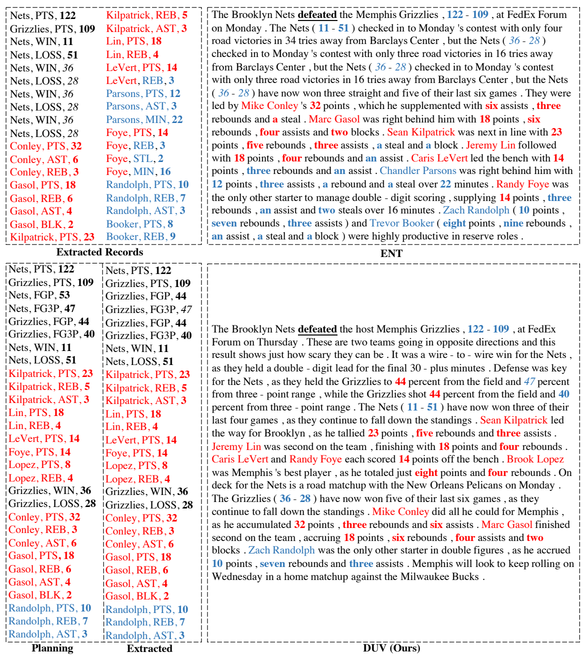
ROTOWIRE v.s. MLB Our model's improvements on CS and CO are significant on ROTOWIRE, but less significant on MLB. Different from ROTOWIRE, MLB additionally provides sequential event data. The two different sources of input can be regards as heterogeneous (Liu et al., 2019). The average statistical data in gold text is 12.69 while event data is 4.16 (extracted by IE model on test set). In Table 4, we discuss CS and CO for two types of data respectively. DU and verification both improve over base model, with verification contributing more overall. They consistently improve on CS F1% and CO on statistical data, but the high CS of base model indicates little room for improvement. Meanwhile, event data is the bottleneck and the drop on that also attributes to the not so significant overall CS and CO improvement. It reveals potential for content planning on heterogeneous input on MLB as future work.

4.4 Case Study

Compared with NCP and gold text in Fig. 1, DUV (Fig. 4) has nice properties: (1) It accurately states that "Nets" with higher points defeated "Grizzlies" while NCP fails. This is due to our model's ability to compare value; (2) Our model can better filter unimportant records (CS P%) while cover the important ones (CS R%) than both NCP and ENT. Note that our model covers all important players

MLB	CS P%			CS R%			CS F1%			CO		
	OA	STAT	EVNT	OA	STAT	EVNT	OA	STAT	EVNT	OA	STAT	EVNT
NCP	**70.89**	**71.04**	70.19	70.45	76.31	52.53	70.67	73.58	60.09	36.08	43.70	37.91
S-NCP	69.73	69.64	70.15	71.60	77.66	53.05	70.65	73.43	**60.41**	36.08	43.25	39.06
S-N+V	69.77	69.65	70.32	72.05	78.79	51.39	70.89	73.94	59.38	36.30	43.71	38.42
DU	69.48	69.48	69.48	72.26	78.45	**53.29**	70.84	73.69	60.32	36.43	43.66	**39.29**
DUV	70.16	70.07	**70.59**	72.31	78.97	51.90	**71.22**	**74.25**	59.82	**36.54**	**44.07**	38.57
-EI	69.84	69.71	70.46	72.32	78.90	52.17	71.06	74.02	59.95	36.15	43.98	38.69
-ER	69.94	69.86	70.30	72.30	78.69	52.74	71.10	74.01	60.27	36.22	43.89	38.50
-RI	69.76	69.70	70.07	**72.41**	**79.01**	52.18	71.06	74.06	59.82	36.29	43.99	38.12
-RR	69.95	69.91	70.15	72.22	78.71	52.36	71.07	74.05	59.96	36.23	43.91	38.54
-RO	69.76	69.65	70.23	**72.41**	78.75	52.98	71.06	73.92	60.40	36.13	43.81	38.31

Table 4: Results of Stage 1 performance on metrics about content planning on MLB's development set. We study overall (OA) results with results for statistical (STAT) data only and event (EVNT) data only.



Extracted Records

Nets, PTS, **122**	Kilpatrick, REB, **5**
Grizzlies, PTS, **109**	Kilpatrick, AST, **3**
Nets, WIN, **11**	Lin, PTS, **18**
Nets, LOSS, **51**	Lin, REB, **4**
Nets, WIN, *36*	LeVert, PTS, **14**
Nets, LOSS, *28*	LeVert, REB, **3**
Nets, WIN, *36*	Parsons, PTS, **12**
Nets, LOSS, *28*	Parsons, AST, **3**
Nets, WIN, *36*	Parsons, MIN, **22**
Nets, LOSS, *28*	Foye, PTS, **14**
Conley, PTS, **32**	Foye, REB, **3**
Conley, AST, **6**	Foye, STL, **2**
Conley, REB, **3**	Foye, MIN, **16**
Gasol, PTS, **18**	Randolph, PTS, **10**
Gasol, REB, **6**	Randolph, REB, **7**
Gasol, AST, **4**	Randolph, AST, **3**
Gasol, BLK, **2**	Booker, PTS, **8**
Kilpatrick, PTS, **23**	Booker, REB, **9**

ENT

The Brooklyn Nets **defeated** the Memphis Grizzlies , **122** - **109** , at FedEx Forum on Monday . The Nets (**11** - **51**) checked in to Monday 's contest with only four road victories in 34 tries away from Barclays Center , but the Nets (*36* - *28*) checked in to Monday 's contest with only three road victories in 16 tries away from Barclays Center , but the Nets (*36* - *28*) checked in to Monday 's contest with only three road victories in 16 tries away from Barclays Center , but the Nets (*36* - *28*) have now won three straight and five of their last six games . They were led by Mike Conley 's **32** points , which he supplemented with **six** assists , **three** rebounds and **a** steal . Marc Gasol was right behind him with **18** points , **six** rebounds , **four** assists and **two** blocks . Sean Kilpatrick was next in line with **23** points , **five** rebounds , **three** assists , **a** steal and **a** block . Jeremy Lin followed with **18** points , **four** rebounds and **an** assist . Caris LeVert led the bench with **14** points , **three** rebounds and **an** assist . Chandler Parsons was right behind him with **12** points , **three** assists , **a** rebound and **a** steal over **22** minutes . Randy Foye was the only other starter to manage double - digit scoring , supplying **14** points , **three** rebounds , **an** assist and **two** steals over 16 minutes . Zach Randolph (**10** points , **seven** rebounds , **three** assists) and Trevor Booker (**eight** points , **nine** rebounds , **an** assist , **a** steal and **a** block) were highly productive in reserve roles .

Planning	Extracted
Nets, PTS, **122**	Nets, PTS, **122**
Grizzlies, PTS, **109**	Grizzlies, PTS, **109**
Nets, FGP, **53**	Grizzlies, FGP, **44**
Nets, FG3P, **47**	Grizzlies, FG3P, *47*
Grizzlies, FGP, **44**	Grizzlies, FGP, **44**
Grizzlies, FG3P, **40**	Grizzlies, FG3P, **40**
Nets, WIN, **11**	Nets, WIN, **11**
Nets, LOSS, **51**	Nets, LOSS, **51**
Kilpatrick, PTS, **23**	Kilpatrick, PTS, **23**
Kilpatrick, REB, **5**	Kilpatrick, REB, **5**
Kilpatrick, AST, **3**	Kilpatrick, AST, **3**
Lin, PTS, **18**	Lin, PTS, **18**
Lin, REB, **4**	Lin, REB, **4**
LeVert, PTS, **14**	LeVert, PTS, **14**
Foye, PTS, **14**	Foye, PTS, **14**
Lopez, PTS, **8**	Lopez, PTS, **8**
Lopez, REB, **4**	Lopez, REB, **4**
Grizzlies, WIN, **36**	Grizzlies, WIN, **36**
Grizzlies, LOSS, **28**	Grizzlies, LOSS, **28**
Conley, PTS, **32**	Conley, PTS, **32**
Conley, REB, **3**	Conley, REB, **3**
Conley, AST, **6**	Conley, AST, **6**
Gasol, PTS, **18**	Gasol, PTS, **18**
Gasol, REB, **6**	Gasol, REB, **6**
Gasol, AST, **4**	Gasol, AST, **4**
Gasol, BLK, **2**	Gasol, BLK, **2**
Randolph, PTS, **10**	Randolph, PTS, **10**
Randolph, REB, **7**	Randolph, REB, **7**
Randolph, AST, **3**	Randolph, AST, **3**

DUV (Ours)

The Brooklyn Nets **defeated** the host Memphis Grizzlies , **122** - **109** , at FedEx Forum on Thursday . These are two teams going in opposite directions and this result shows just how scary they can be . It was a wire - to - wire win for the Nets , as they held a double - digit lead for the final 30 - plus minutes . Defense was key for the Nets , as they held the Grizzlies to **44** percent from the field and *47* percent from three - point range , while the Grizzlies shot **44** percent from the field and **40** percent from three - point range . The Nets (**11** - **51**) have now won three of their last four games , as they continue to fall down the standings . Sean Kilpatrick led the way for Brooklyn , as he tallied **23** points , **five** rebounds and **three** assists . Jeremy Lin was second on the team , finishing with **18** points and **four** rebounds . Caris LeVert and Randy Foye each scored **14** points off the bench . Brook Lopez was Memphis 's best player , as he totaled just **eight** points and **four** rebounds . On deck for the Nets is a road matchup with the New Orleans Pelicans on Monday . The Grizzlies (**36** - **28**) have now won five of their last six games , as they continue to fall down the standings . Mike Conley did all he could for Memphis , as he accumulated **32** points , **three** rebounds and **six** assists . Marc Gasol finished second on the team , accruing **18** points , **six** rebounds , **four** assists and **two** blocks . Zach Randolph was the only other starter in double figures , as he accrued **10** points , **seven** rebounds and **three** assists . Memphis will look to keep rolling on Wednesday in a home matchup against the Milwaukee Bucks .

Figure 4: Generation examples based on tables in Fig. 1. Important/unimportant entities and records are in red/blue. Text that accurately/incorrectly reflects the statistics in table is in bold/italic. Due to page limit, we include generation example on MLB in Appendix.

and their records in this case while only mention one not so impressive player's records; (3) By comparing the content planning (Stage 1) results and actual records mentioned in our model's text (Stage 2), the main challenge indeed lies in the content planning since surface realization can faithfully deliver most information (93.10%) in the same order.

5 Related Work

In the past few years, table-to-text generation has attracted many attentions. To improve text fidelity, Li and Wan (2018) propose to generate templates and then fill the slots, while Nie et al. (2018) use pre-executed operations. However, our work mainly focuses on improving the content planning. Puduppully et al. (2019b) propose to specifically model entities when decoding texts. Different from them, we model numerical values during encoding. Iso et al. (2019) incorporate writers' information to generate text step-by-step. Our work can also consider such information in surface realization (Stage 2). For a fair comparison of all methods, we do not include the use of this model here. Gong et al. (2019) utilize hierarchical encoders with dual attention to consider both the table structure and history information. In terms of building numerical value representations, Spithourakis and Riedel (2018) explore number prediction for language models while Naik et al. (2019) explore numerical embeddings to capture the numeration and magnitude properties of numbers. In our task, generation models rely heavily on copy mechanism to cover numerical values in text and achieve good results. Thus, how to understand numerical values to select records becomes important and we propose to understand them through their context.

6 Conclusion

In order to enhance neural content planning for table-to-text generation, we proposed (1) contextual numerical value representations to help model understand data values and (2) effective rewards to verify a model's inferred important records during training. Experimental results show that our model outperforms competitive baselines in terms of content planning. In the future, we would like to explore enhancement on surface realization jointly to generate better text.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Xiaocheng Feng is the corresponding author of this work. This work is supported by the National Key R&D Program of China via grant 2018YFB1005103 and National Natural Science Foundation of China (NSFC) via grant 61906053 and 61632011.

References

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

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979.

Dimitra Gkatzia. 2016. Content selection in data-to-text systems: A survey. *arXiv preprint arXiv:1610.08375*.

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 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, pages 145–150.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1044–1055.

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. *AAAI*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Kathleen R McKeown. 1985. Text generation: using discourse strategies and focus constraints to generate natural language text.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the 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 Conference on Empirical Methods in Natural Language Processing*, pages 3879–3889.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with content selection and planning. *Proceedings of AAAI Conference on Artificial Intelligence*.

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*.

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

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Richard S Sutton and Andrew G Barto. 1998. Introduction to reinforcement learning, volume 135.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.