Revisiting Challenges in Data-to-Text Generation with Fact Grounding

Hongmin Wang
University of California Santa Barbara
hongmin_wang@cs.ucsb.edu

Abstract

Data-to-text generation models face challenges in ensuring data fidelity by referring to the correct input source. To inspire studies in this area, Wiseman et al. (2017) introduced the RotoWire corpus on generating NBA game summaries from the box- and line-score tables. However, limited attempts have been made in this direction and the challenges remain. We observe a prominent bottleneck in the corpus where only about 60% of the summary contents can be grounded to the boxscore records. Such information deficiency tends to misguide a conditioned language model to produce unconditioned random facts and thus leads to factual hallucinations. In this work, we restore the information balance and revamp this task to focus on fact-grounded data-to-text generation. We introduce a purified and larger-scale dataset, RotoWire-FG (Fact-Grounding), with 50% more data from the year 2017-19 and enriched input tables, hoping to attract more research focuses in this direction. Moreover, we achieve improved data fidelity over the state-of-the-art models by integrating a new form of table reconstruction as an auxiliary task to boost the generation quality.

1 Introduction

Data-to-text generation aims at automatically producing descriptive natural language texts to convey the messages embodied in structured data formats, such as database records (Chisholm et al., 2017), knowledge graphs (Gardent et al., 2017a), and tables (Lebret et al., 2016; Wiseman et al., 2017). Table 1 shows an example from the RotoWire[1] (RW) corpus illustrating the task of generating document-level NBA basketball game summaries from the large box- and line-score tables[2]. It poses great challenges, requiring capabilities to select *what to say* (content selection) from two levels: what entity and which attribute, and to determine *how to say* on both discourse (content planning) and token (surface realization) levels.

Although this excellent resource has received great research attention, very few works (Li and Wan, 2018; Puduppully et al., 2019a,b; Iso et al., 2019) have attempted to tackle the challenges on ensuring data fidelity. This intrigues us to investigate the reason behind and we identify a major culprit undermining researchers' interests: the ungrounded contents in the human-written summaries impedes a model to learn to generate accurate fact-grounded statements and leads to possibly misleading evaluation results when the models are compared against each other.

Specifically, we observe that about 40% of the game summary contents cannot be directly mapped to any input boxscore records, as exemplified by Table 1. Written by professional sports journalists, these statements incorporate domain expertise and background knowledge consolidated from heterogeneous sources that are often hard to trace. The resulting information imbalance hinders a model to produce texts fully conditioned on the inputs and the uncontrolled randomness causes factual hallucinations, especially for the modern encoder-decoder framework (Sutskever et al., 2014; Cho et al., 2014). However, data fidelity is crucial for data-to-text generation besides fluency. In this real-world application, mistaken statements are detrimental to the document quality no matter how human-like they appear to be.

Apart from the popular BLEU (Papineni et al., 2002) metric for text generation, Wiseman et al.

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

[2] Box- and line-score tables contain player and team statistics respectively. For simplicity, we call the combined input the *boxscore table* unless otherwise specified.

Proceedings of The 12th International Conference on Natural Language Generation, pages 311–322,
Tokyo, Japan, 28 Oct - 1 Nov, 2019. ©2019 Association for Computational Linguistics

TEAM	WIN	LOSS	PTS	FG_PCT	BLK	...
Rockets	18	5	108	44	7	
Nuggets	10	13	96	38	7	

PLAYER	H/A	PTS	RB	AST	MIN	...
James Harden	H	24	10	10	38	...
Dwight Howard	H	26	13	2	30	...
JJ Hickson	A	14	10	2	22	...

Column names :
H/A: home/away, PTS: points, RB: rebounds,
AST: assists, MIN: minutes, BLK: blocks,
FG_PCT: field goals percentage

An example hallucinated statement :
After going into halftime down by eight , the Rockets came out firing in the third quarter and out - scored the Nuggets 59 - 42 to seal the victory on the road

The **Houston Rockets** (**18**-**5**) defeated the **Denver Nuggets** (**10**-**13**) **108**-**96** on Saturday. **Houston** has won **2 straight games** and **6** of their last **7**. **Dwight Howard** returned to action Saturday after missing the Rockets ' last **11 games** with a knee injury. He was supposed to be limited to **24 minutes** in the game, but **Dwight Howard** persevered to play **30 minutes** and put up a monstrous double-double of **26 points** and **13 rebounds**. Joining **Dwight Howard** in on the fun was **James Harden** with a triple-double of **24 points**, **10 rebounds** and **10 assists** in **38 minutes**. The **Rockets** ' formidable defense held the **Nuggets** to just **38 percent** shooting from the field. **Houston** will face the **Nuggets** again in their next game, going on the road to **Denver** for their game on Wednesday. **Denver** has lost **4** of their last **5 games** as they struggle to find footing during a tough part of their schedule ... **Denver** will begin a **4 - game** homestead hosting the San Antonio Spurs on Sunday.

Table 1: An example from the RotoWire corpus. Partial box- and line-score tables are on the top left. Grounded entities and numerical facts are in **bold**. Yellow sentences contain **red** ungrounded numerical facts, and team game schedule related statements. A system-generated statement with multiple hallucinations on the bottom left.

(2017) also formalized a set of post-hoc information extraction (IE) based evaluations to assess the data fidelity. Using the boxscore table schema, a sequence of (*entity*, *value*, *type*) records mentioned in a system-generated summary are extracted as the content plan. They are then validated for accuracy against the boxscore table and similarity with the one extracted from the human-written summary. However, any hallucinated facts may unrealistically boost the BLEU score while not penalized by the data fidelity metrics since no records can be identified from the ungrounded contents. Thus the possibly misleading evaluation results inhibit systems to demonstrate excellence on this task.

These two aspects potentially undermine people's interests in this data fidelity oriented table-to-text generation task. Therefore, in this work, we revamp the task emphasizing this core aspect to further enable research in this direction. First, we restore the information balance by trimming the summaries of ungrounded contents and replenish the boxscore table to compensate for missing inputs. This requires the non-trivial extraction of the latent gold standard content plans with high-quality. Thus, we take the efforts to design sophisticated heuristics and achieved an estimated 98% precision and 95% recall of the true content plans, retaining 74% of numerical words in the summaries. This yields better content plans as compared to the 94% precision, 80% recall by Puduppully et al. (2019b) and 60% retainment by Wiseman et al. (2017) respectively. Guided by the high-quality content plans, only fact-grounded contents are identified and retained as shown in Table 1. Furthermore, by expending with 50% more games between the years 2017-19, we obtain the more focused RotoWire-FG (RW-FG) dataset.

This leads to more accurate evaluations and collectively paves the way for future works by providing a more user-friendly alternative. With this refurbished setup, the existing models are then re-assessed on their abilities to ensure data fidelity. We discover that by only purifying the RW dataset, the models can generate more precise facts without sacrificing fluency. Furthermore, we propose a new form of table reconstruction as an auxiliary task to improve fact grounding. By incorporating it into the state-of-the-art Neural Content Planning (NCP) (Puduppully et al., 2019a) model, we established a benchmark on the RW-FG dataset with a 24.41 BLEU score and 95.7% factual accuracy.

Finally, these insights lead us to summarize several fine-grained future challenges based on concrete examples, regarding factual accuracy and intra- and inter- sentence coherence.

Our contributions include:

1. We introduce a purified, enlarged and enriched new dataset to support the more focused fact-grounded table-to-text generation task. We provide high-quality summary facts to table records mappings (content plan) and a more user-friendly experimental setup. All codes and data are freely available[3].

2. We re-investigate existing methods with more insights, establish a new benchmark on this task, and uncover more fine-grained challenges to encourage future research.

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

Type	His	Sch	Agg	Game	Inf
Count	69	33	9	23	23
Percent	43.9	21.0	5.7	14.7	14.7

Table 2: Types of ungrounded contents about statistics related to *His*: history (e.g. recent-game/career high/average) *Sch*: team schedule (e.g. what is next game); *Agg*: aggregation of statistics from multiple players (e.g. the duo of two stars combined scoring ...) ; *Game*: during the game (e.g. a game winning shot with 1 second left); *Inf*: inferred from aggregations (e.g. a player carried the team for winning)

2 Data-to-Text Dataset

This task requires models to take as inputs the NBA basketball game boxscore tables containing hundreds of records and generate the corresponding game summaries. A table can be view as a set of (*entity*, *value*, *type*) records where *entity* is the row name and *type* is the column name in Table 1. **Formally**: Let $\mathbb{E} = \{e_k\}_{k=1}^{K}$ be the set of entities for a game. $\mathbb{S} = \{r_j\}_{j=1}^{S}$ be the set of records where each r_j has a value r_j^m, an entity name r_j^e, a record type r_j^t and r_j^h indicating if the entity is the HOME or AWAY team. For example, a record has r_j^t = POINTS, r_j^e = Dwight Howard, r_j^m = 26, and r_j^h = HOME. The summary has T words: $\hat{y}_{1:T} = \hat{y}_1, \ldots, \hat{y}_T$. A sample is a ($\mathbb{S}$, $\hat{y}_{1:T}$) pair.

2.1 Looking into the RotoWire Corpus

To better understand what kind of ungrounded contents are causing the interference, we manually examine a set of 30 randomly picked samples[4] and categorize the sentences into 5 types whose counts and percentages are tabulated in Table 2.

The *His* type occupies the majority portion, followed by the game-specific *Game*, *Inf*, and *Agg* types, and the remaining goes to *Sch*. Specifically, the *His* and *Agg* types come from exponentially large number of possible combinations of game statistics, and the *Inf* type is based on subjective judgments. Thus, it is difficult to trace and aggregate the heterogeneous sources of origin for such statements to fully balance the input and output. The *Sch* and *Game* types require a sample from a large pool of non-numerical and time-related information, whose exclusion would not affect the nature of the fact-grounding generation task. On the other hand, these ungrounded contents misguide a system to generate hallucinated facts and thus defeat the purpose of developing and evaluating models for fact-grounded table-to-text generation. Thus, we emphasize on this core aspect of the task by trimming contents not licensed by the boxscore table, which we show later still encompasses many fine-grained challenges awaiting to be resolved. While fully restoring all desired inputs is also an interesting research challenge, it is orthogonal to our focus and thus left for future explorations.

2.2 RotoWire-FG

Motivated by these observations, we perform purification and augmentation on the original dataset to obtain the new RW-FG dataset.

2.2.1 Dataset Purification

Purifying Contents: We aim to retain game summary contents with facts licensed by the boxscore records. The sports game summary genre is more descriptive than analytical and aims to concisely cover salient player or team statistics. Correspondingly, a summary often finishes describing one entity before shifting to the next. This fashion of topic shift allows us to identify the topic boundaries using sentences as units, and thus greatly narrows down the candidate boxscore records to be aligned with a fact. The mappings can then be identified using simple pattern-based matching, as also explored by Wiseman et al. (2017). It also enables resolving co-reference by mapping the singular and plural pronouns to the most recently mentioned players and teams respectively. A numerical value associated with an entity is licensed by the boxscore table if it equals to the record value of the desired type. Thus we design a set of heuristics to determine the types, such as mapping “Channing Frye furnished 12 points” to the (*Channing Frye*, *12*, *POINTS*) record in the table. Finally, consecutive sentences describing the same entity is retained if any numerical value is licensed by the boxscore table.

This trimming process introduces negligible influences on the inter-sentence coherence for the summaries. We achieve a 98% precision and a 95% recall of the true content plans and align 74% of all numerical words in the summaries to records in the boxscore tables. The sequence of mapped records is extracted as the content plans and samples describing fewer than 5 records are discarded.

In between the labor-intensive yet imperfect manual annotation and the cheap but inaccu-

[4] For convenience, they are from the validation set and also used later for evaluation purposes.

Versions	Examples	Tokens	Vocab	Types	Avg Len
RW	4.9K	1.6M	11.3K	39	337.1
RW-EX	7.5K	2.5M	12.7K	39	334.3
RW-FG	7.5K	1.5M	8.8K	61	205.9

Table 3: Comparison between datasets. (RW-EX is the enlarged RW with 50% more games)

	Sents	Content Plans	Records	Num-only Records
RW-EX	14.0	27.2	494.2	429.3
RW-FG	8.6	28.5	519.9	478.3

Table 4: Dataset statistics by the average number of each item per sample.

	Quarters				Players	
Sums	1 to 2	1 to 3	2 to 3	2 to 4	bench	starters
	Halves		Quarters			
Diffs	1st	2nd	1	2	3	4

Table 5: Replenished line-score statistics. Each purple cell corresponds to a new record type, defined as applying the the operation in the row names (green) to the source of statistics in the column names (yellow). "Sums" operates on individual teams and "Diffs" is between the two teams. For example, the "1 to 2" cell in the second row means the summation of points scored by a team in the 1st and 2nd "Quarters", the "1st" cell in the fourth row means the difference between the two teams' 1st half points.

rate lexical matching, we achieved better quality through designing the heuristics using similar efforts as training and assembling the IE models by Wiseman et al. (2017). Meanwhile, more accurate content plans provide better reliability during evaluation.

Normalization: To enhance accuracy, we convert all English number words into numerical values. As some percentages are rounded differently between the summaries and the boxscore tables, such discrepancies are rectified. We also perform entity normalization for players and teams, resolving mentions of the same entity to one lexical form. This makes evaluations more user-friendly and less prone to errors.

2.2.2 Dataset Augmentation

Enlargement: Similar to Wiseman et al. (2017), we crawl the game summaries from the *RotoWire Game Recaps*[5] between years 2017-19 and align the summaries with the official *NBA*[6] boxscore tables. This brings 2.6K more games with 56% more tokens, as tabulated in Table 4.

Line-score replenishment: Many team statistics in the summaries are missing in the line-score tables. We recover them by aggregating other boxscore statistics. For example, the number of shots attempted and made by the team for field goals, 3-pointers, and free-throws are calculated by summing their player statistics. Besides, we supplement a set of team point breakdowns as shown in Table 5. The replenishment boosts the recall on numerical values from 72% to 74% and augments the content plans by 1.3 records per sample.

Finalize: We conduct the same purification procedures described in section 2.2.1 after the augmentations. More data collection details are included in Appendix A.

3 Re-assessing Models on Purified RW

3.1 Models

We re-assess three neural network based models on this task[7]. To feed the tables to the models, each record r_j has attribute embeddings for r_j^m, r_j^e, r_j^t, r_j^h and their concatenation is the input.

- ED-CC (Wiseman et al., 2017): This is an Encoder-Decoder (ED) (Sutskever et al., 2014; Cho et al., 2014) model with an 1-layer MLP encoder (Yang et al., 2017), and an LSTM (Hochreiter and Schmidhuber, 1997) decoder with the Conditional Copy (CC) mechanism (Gulcehre et al., 2016).

- NCP (Puduppully et al., 2019a): The Neural Content Planning (NCP) model employs a pointer network (Vinyals et al., 2015) to select a subset of records from the boxscore table and sequentially roll them out as the content plan. Then the summary is then generated only from the content plan using the ED-CC model with a Bi-LSTM encoder.

- ENT (Puduppully et al., 2019b): The ENTity memory network (ENT) model extends the ED-CC model with a dynamically updated entity-specific memory module to capture topic shifts in outputs and incorporate it into each decoder step with a hierarchical attention mechanism.

[5] https://www.RotoWire.com/basketball/game-recaps.php

[6] https://stats.nba.com/

[7] Iso et al. (2019) was released after this work was submitted. It also altered the RW-FG dataset for experiments, so the results would not be directly comparable. The method is worth investigation for future works.

3.2 Evaluation

In addition to using BLEU (Papineni et al., 2002) as a reasonable proxy for evaluating the fluency of the generated summary, Wiseman et al. (2017) designed three types of metrics to assess if a summary accurately conveys the desired information.

Extractive Metrics: First, an ordered sequence of (*entity*, *value*, *type*) triples are extracted from the system output summary as the content plan using the same heuristics in section 2.2.1. It is then checked against the table for its accuracy (RG) and the gold content plan to measure how well they match (CS & CO). Specifically, let $cp = \{r_i\}$ and $cp' = \{r'_i\}$ be the gold and system content plan respectively, and $|.|$ denote set cardinality. We calculate the following measures:

- **Content Selection (CS)**:
  - Precision (**CSP**) = $|cp \cap cp'| \,/\, |cp'|$
  - Recall (**CSR**) = $|cp \cap cp'| \,/\, |cp|$
  - F1 (**CSF**) = $2PR/(P+R)$
- **Relation Generation (RG)**:
  - Count(#) = $|cp'|$
  - Precision (**RGP**) = $|cp' \cap \mathbb{S}| \,/\, |cp'|$
- **Content Ordering (CO)**:
  - **DLD**: normalized Damerau Levenshtein Distance (Brill and Moore, 2000) between *cp* and *cp*′

CS and **RG** measures the "what to say" and **CO** measures the "how to say" aspects.

3.3 Experiments

Setup: To re-investigate the existing three methods on the ability to convey accurate information conditioned on the input, we assess them by training on the purified `RW` corpus. To demonstrate the differences brought by the purification process, we keep all other settings unchanged and report results on the original validation and test sets after performing early stopping (Yao et al., 2007) based on the BLEU score.

Results: As shown in Table 6, we observe increase in Relation Generation Precision (RGP) and on-par performance for Content Selection (CS) and Content Ordering (CO). In particular, Relation Generation Precision (RGP) is substantially increased by an average 2.7% for all models. The Content Selection (CS) and Content Ordering (CO) measures fluctuate above and below the references, with the biggest disparity on Content Selection Precision (CSP), Content Selection Recall (CSR) and Content Ordering (CO) for the ENT model. Since output length is a main independent variable for this set of experiments and a crucial factor in BLEU score as well, we report the breakdowns in Table 7. Specifically, the NCP model shows consistent improvements on all BLEU 1-4 scores, similarly for ENT on the validation set. Among all fluctuation around the references, nearly all models demonstrate an increase in BLEU-1 and BLEU-4 precision. Reflected on the BP coefficients, models trained on the purified summaries produces shorter outputs, which is the major reason for lower BLEU scores when using the un-purified summaries as the references.

3.4 How Purification Affects Performance

First, simply replacing with the purified training set leads to considerable improvements in the Relation Generation Precision (RGP). This is because removing the ungrounded facts (e.g. His, Agg, and Game types) alleviates their interference with the model while learning when and where to copy over a correct numerical value from the table. Besides, since the ungrounded facts do not contribute to the gold or system output content plan during the information extraction process, the other extractive metrics Content Selection (CS) and Content Ordering (CO) measures stay on-par.

One abnormality is the big difference in the Content Selection (CS) and Content Ordering (CO) measures from the ENT model. This is not that surprising after examining the outputs, which appear to collapse into template-like summaries. For example, 97.8% sentences start with the game points followed by a pattern "XX were the superior shooters" where XX represents a team. Tracing back to the model design, it is explicitly trained to model topic shifts on the token level during generation, which instead happens more often on the sentence level. As a result, it degenerates to remembering a frequent discourse-level pattern from the training data. We observe a similar pattern on the outputs from original outputs by Puduppully et al. (2019b), which is aggravated

[8] For fair comparison, we report results of ENT model after fixing a bug in the evaluation script as endorsed by the author of Wiseman et al. (2017) at https://github.com/harvardnlp/data2text/issues/6

Model	Dev						Test					
	RG		CS			CO	RG		CS			CO
	#	P%	P%	R%	F1%	DLD%	#	P%	P%	R%	F1%	DLD%
ED-CC	23.95	75.10	28.11	35.86	31.52	15.33	23.72	74.80	29.49	36.18	32.49	15.42
ED-CC(FG)	22.65	78.63	29.48	34.08	31.61	14.58	23.36	79.88	29.36	33.36	31.23	13.87
NCP	33.88	87.51	33.52	51.21	40.52	18.57	34.28	87.47	34.18	51.22	41.00	18.58
NCP(FG)	31.90	90.20	34.53	49.74	40.76	18.29	33.51	91.46	33.96	49.14	40.16	18.16
ENT[8]	21.49	91.17	40.50	37.78	39.09	19.10	21.53	91.87	42.61	38.31	40.34	19.50
ENT(FG)	30.08	93.74	30.43	48.64	37.44	16.53	30.66	93.09	32.40	41.69	36.46	16.44

Table 6: Comparison between models trained on RW and RW-FG

Model	Dev						Test					
	B1	B2	B3	B4	BP	BLEU	B1	B2	B3	B4	BP	BLEU
ED-CC	44.42	**18.16**	**9.40**	5.95	**1.00**	**14.57**	43.22	**17.64**	**9.16**	5.81	**1.00**	**14.19**
ED-CC(FG)	**46.61**	17.70	9.33	**6.21**	0.59	8.74	**45.75**	17.14	9.05	**5.98**	0.61	8.68
NCP	48.95	20.58	10.70	6.96	**1.00**	**16.19**	49.77	21.19	11.31	7.46	**0.96**	**16.50**
NCP(FG)	**56.63**	**24.15**	**12.45**	**8.13**	0.54	10.45	**56.33**	**23.92**	**12.42**	**8.11**	0.53	10.25
ENT	51.57	21.92	11.87	8.08	**0.88**	**15.97**	53.23	**23.07**	**12.78**	**8.78**	**0.84**	**16.12**
ENT(FG)	**56.08**	**23.29**	**12.29**	**8.16**	0.44	8.92	**55.03**	21.86	11.38	7.38	0.57	10.17

Table 7: Breakdown of BLEU scores for models trained on RW and RW-FG

when trained on the purified dataset. On the other hand, the NCP model decouples the content selection and planning on the discourse level from the surface realization on the token level, and thus generalizes better.

4 A New Benchmark on RW-FG

With more insights about the existing methods, we take a step further to achieve better data fidelity. Wiseman et al. (2017) achieved improvements on the ED with Joint Copy (JC) (Gu et al., 2016) model by introducing an reconstruction loss (Tu et al., 2017) during training. Specifically, the decoder states at each time step are used to predict record values in the table to enable broader input information coverage.

However, we take a different point of view: one key mechanism to avoid reference errors is to ensure that the set of numerical values mentioned in a sentence belongs to the correct entity with the correct record field type. While the ED-CC model is trained to achieve such alignments, it should also be able to accurately fill the numbers back to the correct cells in an empty table. This should be done by only accessing the column and row information of the cells without explicitly knowing the original cell values. Further leveraging on the planner output of the NCP model, the candidate cells to be filled can be reduced to the content plan cells selected by the planner. With this intuition, we devise a new form of table reconstruction (TR) task incorporated into the NCP model.

Specifically, each content plan record has attribute embeddings for r_j^e, r_j^t, and r_j^h, excluding its value, and we encode them using a 1-layer MLP (Yang et al., 2017). We then employ the Luong et al. (2015) attention mechanism at each $\hat{y}_t$ if it is a numerical value with the encoded content plan as the memory bank. The attention weights are then viewed as probabilities of selecting each cell to fill the number $\hat{y}_t$. The model is additionally trained to minimize the negative log-likelihood of the correct cell.

4.1 Experiments

Setup: We assess models on the RW-FG corpus to establish a new benchmark. Following Wiseman et al. (2017), we split all samples into train (70%), validation (15%), and test (15%) sets, and perform early stopping (Yao et al., 2007) using BLEU (Papineni et al., 2002). We adapt the template-based generator by Wiseman et al. (2017) and remove the ungrounded end sentence since they are eliminated in RW-FG.

Results: As shown in Table 8, the template model can ensure high Relation Generation Precision (RGP) but is inflexible as shown by other measures. Different from Puduppully et al. (2019b), the NCP model is superior on all measures among the baseline neural models. The ENT model only outperforms the basic ED-CC model but surprisingly yields lower Content Selection (CS) measures. Our NCP+TR model outperforms all baselines except for slightly lower Content Selection Precision (CSP) compared to the NCP model.

Model	Dev							Test						
	RG		CS			CO	BLEU	RG		CS			CO	BLEU
	#	P%	P%	R%	F1%	DLD%		#	P%	P%	R%	F1%	DLD%	
TMPL	51.81	99.09	23.78	43.75	30.81	10.06	11.91	51.80	98.89	23.98	43.96	31.03	10.25	12.09
WS17	30.47	81.51	36.15	39.12	37.57	18.56	21.31	30.28	82.16	35.84	38.40	37.08	18.45	20.80
ENT	35.56	93.30	40.19	50.71	44.84	17.81	21.67	35.69	93.72	39.04	49.29	43.57	17.50	21.23
NCP	36.28	94.27	**43.31**	55.96	48.91	24.08	24.49	35.99	94.21	**43.31**	55.15	48.52	23.46	23.86
NCP+TR	**37.04**	**95.65**	43.09	**57.24**	**49.17**	**24.75**	**24.80**	**37.49**	**95.70**	42.90	**56.91**	**48.92**	**24.47**	**24.41**

Table 8: Performances of models on RW-FG

Model	Total(#)	RP(%)	WC(%)	UG(%)	IC(%)
NCP	246	9.21	11.84	3.07	5.26
NCP+TR	228	3.66	8.94	3.25	2.03

Table 9: Error types of manual evaluation. *Total*: number of sentences; *RP*: Repetition; *WC*: Wrong Claim; *UG*: Ungrounded sentence; *IC*: Incoherent sentence

.

4.2 Discussion

We observe that the ED-CC model produces the least number of candidate records, and correspondingly achieves the lowest Content Selection Recall (CSR) compared to the gold standard content plans. As discussed in section 3.4, the template-like discourse pattern produced by the ENT model noticeably deteriorates its performance. It is completely outperformed by the NCP model and even achieves lower CO-DLD than the ED-CC model. Finally, as supported by the extractive evaluation metrics, employing table reconstruction as an auxiliary task indeed boosts the decoder to produce more accurate factual statements. We discuss in more detail as follows.

4.2.1 Manual Evaluation

To gain more insights into how exactly NCP+TR improves from NCP in terms of factual accuracy, we manually examined the outputs on the 30 samples. We compare the two systems after categorizing the errors into 4 types. As shown in Table 9, the largest improvement comes from reducing repeated statements and wrong fact claims, where the latter involves referring to the wrong entity or making the wrong judgment of the numerical value. The NCP+TR generally produces more concise outputs with a reduction in repetitions, consistent with the objective for table reconstruction.

4.2.2 Case study

Table 10 shows a pair of outputs by the two systems. In this example, the NCP+TR model can correct wrong the player name "*Jahlil Okafor*" by "*Joel Embiid*", while keeping the statistics intact. It also avoids repeating on "*Channing Frye*" and the semantically incoherent expression about "*Kevin Love*" and "*Kyrie Irving*". Nonetheless, this NCP output selects more records to describe the progress of the game. This shows how the NCP+TR trained with more constraints behaves more accurately but conservatively.

5 Errors and Challenges

Having revamped the task with better focus, reassessed existing and improved models, we discuss 3 future directions in this task with concrete examples in Table 11:

Content Selection: Since writers are subjective in choosing *what to say* given the boxscore, it is unrealistic to force a model to mimic all kinds of styles. However, a model still needs to learn from training to select both the salient (e.g. surprisingly high/low statistics for a team/player) and the popular (e.g. the big stars) statistics. One potential direction is to involve multiple human references to help reveal such saliency and make Content Ordering (CO) and Content Selection (CS) measures more interpretive. This is particularly applicable for the sports domain since a game can be uniquely identified by the teams and date but mapped to articles from different sources. Besides, multi-reference has been explored for evaluating data-to-text generation systems (Novikova et al., 2017) and for content selection and planning (Gehrmann et al., 2018). It has also been studied in machine translation for evaluation (Dreyer and Marcu, 2012) and training (Zheng et al., 2018).

Content Planning: Content plans have been extracted by linearly rolling out the records and topic shifts are modeled as sequential changes between adjacent entities. However, this fashion does not reflect the hierarchical discourse structures of a document and thus ensures neither intra- nor inter-sentence coherence. As shown by the errors in (1)

The **Cleveland Cavaliers** defeated the **Philadelphia 76ers** , **102 - 101** , at **Wells Fargo Center** on Monday evening . **LeBron James** led the way with a **25 - point** , **14 - assist** double double that also included **8 rebounds** , **2 steals** and **1 block** . **Kevin Love** followed with a **20 - point** , **11 - rebound** double double that also included **1 assist** and **1 block** . **Channing Frye** led the bench with **12 points** , **2 rebounds** , **2 assists** and **2 steals** **Kyrie Irving** managed **8 points** , **7 rebounds** , **2 assists** and **2 steals** . ... **Joel Embiid** 's **22 points** led the **Sixers** , a total he supplemented with **6 rebounds** , **2 assists** , **4 blocks** and **1 steal** ...
The **Cleveland Cavaliers** defeated the **Philadelphia 76ers** , **102 - 101** , at **Wells Fargo Center** on Friday evening . The **Cavaliers** came out of the gates hot , jumping out to a **34 - 15** lead after 1 quarter . However , the **Sixers** (**0 - 5**) stormed back in the second to cut the deficit to just **2** points by halftime . However , the light went on for Cleveland at intermission , as they built a **9** - point lead by halftime . **LeBron James** led the way for the Cavaliers with a **25 - point** , **14 - assist** double double that also included **8 rebounds** , **2 steals** and **1 block** . Kyrie Irving followed **Kevin Love** with a **20 - point** , **11 - rebound** double double that also included **1 assist** and **1 block** . **Channing Frye** furnished **12 points** , **2 rebounds** , **2 assists** and **2 steals** ... **Channing Frye** led the bench with **12 points** , **2 rebounds** , **2 assists** and **2 steals** . **Jahlil Okafor** led the **Sixers** with **22 points, 6 rebounds** , **2 assists, 4 blocks and 1 steal** ... **Jahlil Okafor** managed **14 points** , **5 rebounds** , **3 blocks** and **1 steal** .

Table 10: Case study comparing NCP+TR (above) and NCP (below). The records identified are in **bold**. The pair of sentences in orange shows an referring error to **Jahlil Okafor** is corrected above to **Joel Embiid**, where all the trailing statistics actually belong to **Joel Embiid**, and **Jahlil Okafor**'s actual statistics are described at the end. The yellow sentences repeats on the same player. The green sentences actually shows some more contents selected by the NCP model. The blue sentence is a tricky one, where it should describe **Kyrie Irving**'s statistics but actually describing **Kevin Love**'s but the summary above does not have this issue.

(1) Intra-sentence coherence:

- The **Lakers** were the *superior shooters* in this game , going **48 percent** from the field and **24 percent** from the three point line , while the **Jazz** went **47 percent** from the floor and just **30 percent** from beyond the arc.
- The **Rockets** got off to a quick start in this game, *out scoring* the **Nuggets** **21-31** right away in the 1st quarter.

(2) Inter-sentence coherence:

- **LeBron James** was the lone bright spot for the **Cavaliers** , as he led the team with **20 points** . **Kevin Love** was *the only Cleveland starter in double figures* , as he tallied **17 points** , **11 rebounds** and **3 assists** in the loss.
- ***Dirk Nowitzki*** *led the* ***Mavericks*** *in scoring* , finishing with **22 points** (**7 - 13 FG** , **3 - 5 3PT** , **5 - 5 FT**) , **5 rebounds** and **3 assists** in **37 minutes**. *He* 's had a very strong stretch of games , scoring **17 points** on **6** - for - **13** shooting from the field and **5** - for - **10** from the three point line. ***JJ Barea*** *finished with* ***32 points*** (**13 - 21 FG** , **5 - 8 3PT**) and **11 assists** ...

(3) Incorrect claim:

- The **Heat** were able to force **20** turnovers from the **Sixers**, which may have been the difference in this game.

Table 11: Cases for three major types of system errors

in Table 11, the links between entities and their numerical statistics are not strictly monotonic and switching the order results in errors.

On the other hand, autoregressive training for creating such content plans limits the model to capture frequent sequence patterns rather than allowing diverse arrangements. Moryossef et al. (2019) demonstrates isolating the content planning from the joint end-to-end training and employing multiple valid content plans during testing. Although the content plan extraction heuristics are dataset-dependent, it is worth exploring for data in a closed domain like RW.

Surface Realization: Although the NCP+TR model has achieved nearly 96% Relation Generation Precision (RGP), it is still paramount to keep on improving data accuracy since one single mistake is destructive to the whole document. The challenge is more with the evaluation metrics. Specifically, all extractive metrics only validate if an extracted record maps to the true entity and type but disregards the semantics of its contexts. For example (2) in Table 11, even assuming the linear ordering of records, their context still causes inter-sentence incoherence. In particular, both LeBron and Kevin scored double digits and JJ Barea leads the scores rather than Dirk. For another example (3), the 20 turnovers records are selected to be Heat's but expressed falsely as Sixers'. As pointed out by Wiseman et al. (2017), this may require the integration of semantic or reference-based constraints during generation. The number magnitudes should be incorporated. For example, Nie et al. (2018) has devised an interesting idea to implicitly improve coherence by supplementing the input with pre-computed results from algebraic operations on the table. Moreover, Qin et al. (2018) proposed to automatically align the game summary with the record types in the input table on the phrase level. It can potentially be combined with the operation results to correct incoherence errors and improve the generations.

6 Related Works

Various forms of structured data has been used as input for data-to-text generation tasks, such as tree (Belz et al., 2011; Mille et al., 2018), graph (Konstas and Lapata, 2012), dialog moves (Novikova et al., 2017), knowledge base (Gardent et al., 2017b; Chisholm et al., 2017), database (Konstas and Lapata, 2012; Gardent et al., 2017a; Wang et al., 2018), and table (Wiseman et al., 2017; Lebret et al., 2016). The RW corpus we studied is from the sports domain which has attracted great interests (Chen and Mooney, 2008; Mei et al., 2016; Puduppully et al., 2019b). However, unlike generating the one-entity descriptions (Lebret et al., 2016; Wang et al., 2018) or having the output strictly bounded by the inputs (Novikova et al., 2017), this corpus poses additional challenges since the targets contain ungrounded contents. To facilitate better usage and evaluation of this task, we hope to provide a refined alternative, similar to the purpose by Castro Ferreira et al. (2018).

7 Conclusion

In this work, we study the core fact-grounding aspect of the data-to-text generation task and contribute a purified, enlarged, and enriched RotoWire-FG corpus with a more fair and reliable evaluation setup. We re-assess existing models and found that the more focused setting helps the models to express more accurate statements and alleviate fact hallucinations. Improving the state-of-the-art model and setting a benchmark on the new task, we reveal fine-grained unsolved challenges hoping to inspire more research in this direction.

Acknowledgments

Thanks for the generous and valuable feedback from the reviewers. Special thanks to Dr. Jing Huang and Dr. Yun Tang for their unselfish guidance and support.

References

Anja Belz, Mike White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *ENLG*.

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.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 171–176, Tilburg University, The Netherlands. Association for Computational Linguistics.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 128–135.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111.

Markus Dreyer and Daniel Marcu. 2012. HyTER: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171, Montréal, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The webnlg challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 124–133.

Sebastian Gehrmann, Falcon Z. Dai, Henry Elder, and Alexander M. Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 46–56.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

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)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

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

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*, pages 2102–2113, Florence, Italy. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 369–378.

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.

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, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1044–1055.

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, Lisbon, Portugal. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia. Association for Computational Linguistics.

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.

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.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*.

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, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.

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.

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.

Guanghui Qin, Jin-Ge Yao, Xuening Wang, Jinpeng Wang, and Chin-Yew Lin. 2018. Learning latent semantic annotations for grounding natural language to structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3761–3771, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3097–3103.

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.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, Tilburg University, The Netherlands. 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.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1850–1859, Copenhagen, Denmark. Association for Computational Linguistics.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Renjie Zheng, Mingbo Ma, and Liang Huang. 2018. Multi-reference training with pseudo-references for neural translation and text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3188–3197, Brussels, Belgium. Association for Computational Linguistics.

A Appendices

A.1 Data Collection Details

- We use the text2num[9] package to convert all English number words into numerical values

- We first get the summary title, date, and the contents from *RotoWire Game Recaps*. The title contains the home and visiting team. Together with the date, this game is uniquely identified with a *GAME_ID*. Then we use the *nba_api*[10] package to query the *stats.nba.com* by *NBA.com*[11] to obtain the game boxscore and line scores. Wiseman et al. (2017) used the *nba_py*[12] package , which unfortunately has become obsolete due to lack of maintenance. To obtain the line scores with the same set of column types as the original RotoWire dataset, we collectively used two APIs, *BoxScoreTraditionalV2* and *BoxScoreSummaryV2*.

[9] https://github.com/ghewgill/text2num/blob/master/text2num.py

[10] https://github.com/swar/nba_api

[11] www.nba.com; https://stats.nba.com/

[12] https://github.com/seemethere/nba_py