Plan-then-Generate: Controlled Data-to-Text Generation via Planning

Yixuan Su[♠,*] **David Vandyke**[♡] **Sihui Wang**[♡] **Yimai Fang**[♡] **Nigel Collier**[♠]
[♠]Language Technology Lab, University of Cambridge
[♡]Apple
{ys484,nhc30}@cam.ac.uk
{dvandyke,sihui_wang,yimai_fang}@apple.com

Abstract

Recent developments in neural networks have led to the advance in data-to-text generation. However, the lack of ability of neural models to control the structure of generated output can be limiting in certain real-world applications. In this study, we propose a novel Plan-then-Generate (PlanGen) framework to improve the controllability of neural data-to-text models. Extensive experiments and analyses are conducted on two benchmark datasets, ToTTo and WebNLG. The results show that our model is able to control both the intra-sentence and inter-sentence structure of the generated output. Furthermore, empirical comparisons against previous state-of-the-art methods show that our model improves the generation quality as well as the output diversity as judged by human and automatic evaluations.

1 Introduction

Generating natural language from structured data (Gatt and Krahmer, 2018), i.e. data-to-text generation, is a research problem that is crucial to many downstream NLP applications. Some examples are dialogue systems (Wen et al., 2016), restaurant assistant (Novikova et al., 2017), and open domain question answering (Chen et al., 2021).

To address this task, many researchers have designed sophisticated neural models based on various methods, such as soft-template (Wiseman et al., 2018), copy mechanism (Gehrmann et al., 2018), and pre-trained language models (Kale and Rastogi, 2020; Ribeiro et al., 2020). While achieving impressive results, most existing studies only focused on producing results that are close to the references. On the other hand, the controllability of such models is still under-explored, i.e. what to generate and in what order (the output structure) in their outputs cannot be explicitly controlled by the users.

We argue that the model's ability to control the structure of its output is highly desirable for at least two reasons. (1) Arranging the structure of the output in a certain form enables it to have greater naturalness, as the structure of the sentence often reflects the salience of the entities it contains (Poesio et al., 2004). Suppose we have a digital assistant which replies to user queries based on knowledge tables like Table 1. Then, for a user query "*Who played Evelyn in Kids in Love?*", a natural response is "*Evelyn in Kids in Love was played by Alma Jodorowsky.*". In contrast, to a different query "*What role did Alma Jodorowsky play in Kids in Love?*", a natural response would be "*Alma Jodorowsky played Evelyn in Kids in Love.*". While both answers are semantically equivalent, producing the answer with the most appropriate structure allows the system to sound less robotic and be easily understood. (2) It allows the model to generate outputs with diverse structures by simply changing the input planning information (i.e. a content plan), which could potentially benefit other applications such as paraphrasing and data augmentation.

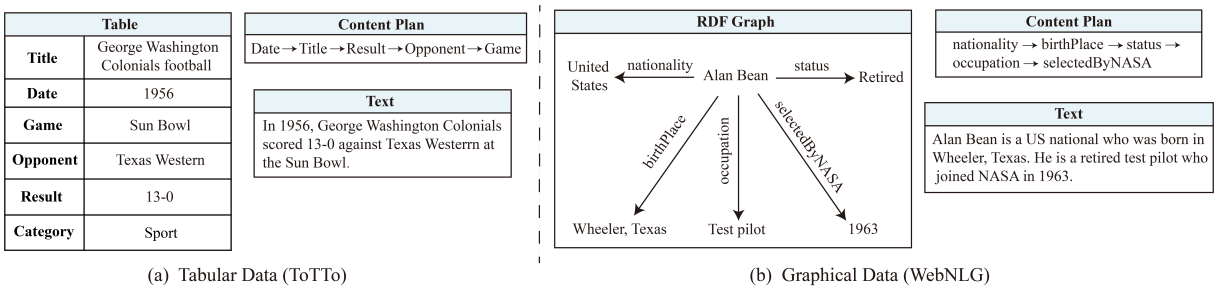
Knowledge Table			
Title	Kids in Love	**Name**	Alma Jodorowsky
Year	2016	**Role**	Evelyn

Table 1: An Example of Knowledge Table

To control the output structure, we need an intermediate "planning" signal (i.e. a content plan) which informs the model what to generate and in what order. To this end, we propose a Plan-then-Generate (PlanGen) framework which consists of two components: a content planner and a sequence generator. Given the input data, the content planner first predicts the most plausible content plan that the output should follow. Then, the sequence generator takes the data and the content plan as input to generate the result. To further ensure the controllability of our model, we propose a structure-aware reinforcement learning (RL) objective that encourages the generated output to adhere to the given

[*]Work done while the author was an intern at Apple.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 895–909
November 7–11, 2021.



(a) Tabular Data (ToTTo) (b) Graphical Data (WebNLG)

Figure 1: Plot illustrating the relationship between the structured data, the content plan, and the reference text for examples from (a) ToTTo dataset (tabular data) and (b) WebNLG dataset (graphical data with RDF structure).

content plan. In this work, we formulate the intermediate content plan as an ordered list of tokens for its simplicity and wide applicability to data with different structures. For tabular data, each token in the content plan is a slot key from the table. As for graphical data with RDF structure, each token represents the predicate from an RDF triple. In Figure 1, we provide examples for both cases.

To fully evaluate our approach, we test the proposed model on two benchmarks with different data structures: (i) ToTTo dataset (Parikh et al., 2020) with tabular data, and (ii) WebNLG dataset (Colin et al., 2016; Gardent et al., 2017) with graphical data. Compared with previous state-of-the-art approaches, our model achieves better performance in terms of generation quality as judged by both human and automatic evaluations. In particular, the results also show that the outputs of our model are highly controllable and contain diverse structures.

In summary, our contributions are: (1) A novel Plan-then-Generate (PlanGen) framework that consists of a content planner and a sequence generator for data-to-text generation. (2) Extensive automatic and human evaluations reporting state-of-the-art results on two benchmark datasets. (3) In-depth analysis revealing the merits of the proposed approach in terms of controllability and diversity.

2 Related Work

Data-to-text generation is a long-standing problem (Reiter and Dale, 1997) that aims at producing natural language descriptions of structured data. Traditional systems are primarily built on template-based algorithms (Oh and Rudnicky, 2000; Stent et al., 2004; Kondadadi et al., 2013). With recent advances in deep learning, researchers have shifted their attention to neural generation models that can be summarized into two categories.

End-to-End Models. Many existing studies are dedicated to building end-to-end neural models with different strategies like soft-templates (Wiseman et al., 2018; Ye et al., 2020), attention awareness (Liu et al., 2018; Colin and Gardent, 2019), and retrieved prototypes (Li et al., 2020; Su et al., 2021b). Gehrmann et al. (2018), Puduppully et al. (2019a,b), and Chen et al. (2020b) adopted copy mechanism for content selection to improve the information coverage of the outputs. With recent advance in pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Lewis et al., 2020), several researchers (Chen et al., 2020a,b; Kale and Rastogi, 2020; Ribeiro et al., 2020) have studied the ways to adapt PLMs into the data-to-text generation task.

Pipeline Models. Another line of research investigates ways to tackle the generation problem in a pipeline framework. Ma et al. (2019) proposed to first use a classifier to select the key contents. The planning and surface realisation of the selected contents are then addressed by a subsequent Seq2seq model. More related to our work, some researchers studied how neural models can benefit from traditional NLG steps (Kukich, 1983; McKeown, 1992), that is, (i) content planning and (ii) surface realisation. To simultaneously select the key contents and arrange their orderings (i.e. content planning), different strategies are proposed such as the most probable traversal of graph trees (Moryossef et al., 2019), the ordering of graph nodes (Zhao et al., 2020), and the multi-step pipeline that includes discourse ordering, lexicalization, and regular expression generation (Ferreira et al., 2019). While achieving satisfactory results, these approaches can only be applied to data with graphical structure. Compared with previous studies, we show that our content planning approach is more accurate and less dependent on the data structure. In addition, by providing the desired content plan, our model can control the output structure on both the intra-sentence and inter-sentence levels (§7.3).

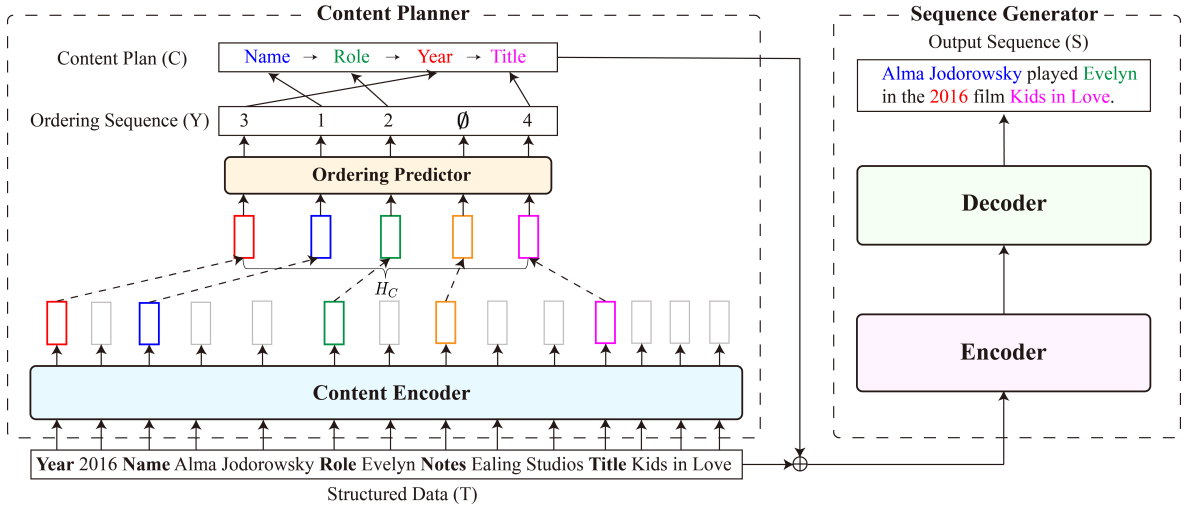


Figure 2: **PlanGen Framework**: Given the structured data (T), a content plan (C) is first predicted by the content planner (left). The sequence generator (right) then takes the structured data and the predicted content plan as input to generate the output (S). Note that, the content plan can also be specified by the user for a controlled generation.

3 Preliminaries

Dataset. In this study, our training dataset is defined as $\mathcal{D} = \{(T, C, S)_i\}_{i=1}^{|D|}$. (1) T is the linearized structured data and it is defined as $T = \{t_1, ..., t_{|T|}\}$. For data with tabular structure, each item $t_i = \{k_i, v_i\}$ is a pair of slot key k_i and slot value v_i (e.g., (*Date*, *1956*) in Figure 1(a)). As for graphical data with RDF structure, each item $t_i = \{s_i, p_i, o_i\}$ represents a RDF triple, where s_i, p_i, and o_i are subject, predicate, and object, respectively. For instance, in Figure 1(b), ("Alan Bean", "status", "Retired") is a RDF triple. (2) The reference content plan C is defined as $C = \{c_1, .., c_{|C|}\}$, where each token c_i either denotes a slot key (for tabular data) or a predicate (for graphical data). The content plan is thus a selection of the content from the structured data that should appear in the output, in a particular order. (3) The $S = \{s_1, .., s_{|S|}\}$ denotes the reference text.

Content Plan Construction. Note that the original ToTTo and WebNLG datasets only consist of pairs of structured data and reference text. Thus, we use a heuristic delexicalizer $\mathcal{F}$ to construct the reference content plan. For a tabular data T, given the reference text S, the content plan $C = \mathcal{F}(T, S)$ is built by replacing the parts of the reference text that comes from the table slot values with the corresponding slot keys. For instance, suppose we have a text "*Alma Jodorowsky played Evelyn in Kids in Love.*" and Table 1, then the resulting content plan is {"Name"→"Role"→"Title"}. For graphical data with RDF structure, we apply a similar procedure to build the reference content plan by replacing the parts of the reference text that comes from the objects of the RDF triples with the corresponding predicates. In Figure 1, we show examples of reference content plan for both cases.

4 Methodology

Figure 2 depicts the proposed Plan-then-Generate (P2G) framework. Given the input data, the content planner (§4.1) first predicts the most probable content plan. The sequence generator (§4.2) then takes the structured data and the predicted content plan to generate the output. In the following, we elaborate the details of the proposed framework.

4.1 Content Planner

Our content planner consists of two components. The first part is a content encoder which takes the data T as input and produces its representation $H_T \in \mathbb{R}^{|T| \times n}$, where n is the output size. We construct our content encoder with a pre-trained BERT-base model (Devlin et al., 2019).

After getting the data representation, we select the hidden states from H_T that corresponds to the tokens[1] that might appear in the content plan. Here, we denote the selected hidden states $H_C \in \mathbb{R}^{|C| \times n}$ as $H_C = \{h_1^c, ..., h_{|C|}^c\}$, where $|C|$ is the number of selected tokens from the input data. Next, H_C is fed into the ordering predictor which predicts the orderings of the selected tokens in the predicted

[1]For tabular data, the selected tokens correspond to all slot keys from the table. Similarly, for graphical data, the selected tokens correspond to the predicates of all input RDF triples.

content plan. Inspired by Su et al. (2021a), we model the ordering predictor as a linear-chain conditional random field (CRF) (Lafferty et al., 2001) for its ability to compute the global optimal ordering sequence. When predicting the ordering, the ordering predictor is allowed to emit an empty label $\emptyset$ which indicates the omission of the corresponding token in the content plan.

During training, the likelihood of the ordering sequence Y defined by the content plan is

$$P_{\text{CRF}}(Y|H_C) = \frac{e^{f(Y,H_C)}}{\sum_{Y'} e^{f(Y',H_C)}} = \frac{1}{Z}\exp(\sum_{i=1}^{|C|}\Phi_{y_i}(h_i^c) + \sum_{i=2}^{|C|} M_{y_{i-1},y_i}). \quad (1)$$

Here, $\Phi_{y_i}(h_i^c)$ is the label score of y_i at step i, where label y_i indicates the position of the token in the final content plan. Taking Figure 2 as an example, the position of the "Name" key is 1, meaning that "Name" should appear in the front of the content plan. By predicting the positions instead of the actual slot keys, at test time, our model can handle tables with out-of-vocabulary slot keys that did not appear in the training set. In practice, Φ is parameterized by a feed-forward layer. The M_{y_{i-1},y_i} denotes the transition score from position y_{i-1} to position y_i, and M is a learnable transition matrix.

During inference, the ordering sequence is predicted as $\tilde{Y}$ as $\tilde{Y} = \arg\max_{Y'} P_{\text{CRF}}(Y'|H_C)$. As shown in the example of Figure 2, given all the slot keys {"Year", "Name", "Role", "Notes", "Title"} from the table, the predicted ordering sequence is {3, 1, 2, $\emptyset$, 4}. The content plan {"Name" → "Role" → "Year" → "Title"} can then be predicted by omitting the "Notes" key and re-arranging other keys following the predicted ordering sequence.

4.2 Sequence Generator

Our sequence generator is built on a BART-base model (Lewis et al., 2020) which consists of a transformer based encoder-decoder architecture.

Given the structured data T, the reference content plan C, and the reference text S, the learning objective of the sequence generator is defined as

$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{|S|}\log P_G(S_i|S_{<i}; E([T:C])), \quad (2)$$

where E, G are the encoder and decoder, and $[\cdot:\cdot]$ denotes the concatenation operation.

4.3 Structure-Aware RL Training

We note that the structure of the generated sequence can only be accurately measured on the sequence-level, which is not directly optimized by the token-level objective (Eq. (2)). Therefore, to encourage the generator to follow the sequence-level structure defined by the content plan, we incorporate reinforcement learning into our training process.

Formally, in training, given the structured data T and the reference content plan C, the generator first samples an output sequence $S' = (S'_1, ..., S'_{|S'|})$, where S'_t is the token sampled at time step t. The generator parameters θ are then updated using the REINFORCE algorithm (Williams, 1992) as

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{S'\sim P_\theta(T,C)}[R(S,S',T,C)] = \quad (3)$$
$$-R(S,S',T,C)\sum_{i=1}^{|S'|}\log P_G(S'_i|S'_{<i}; E([T:C])).$$

The reward function $R(S,S',T,C)$ measures the structure of the sampled sequence S' against the input content plan C, and its surface form against the reference text S as

$$R(S,S',T,C) = B(S,S') + B(C,C'), \quad (4)$$

where $B(\cdot,\cdot)$ is the BLEU score (Papineni et al., 2002). $C' = \mathcal{F}(T,S')$, and $\mathcal{F}$ is described in §3. By optimizing Eq. (3), the structure of the output is encouraged to follow the content plan.

4.4 Learning

The learning objective of the content planner is $\mathcal{L}_{\text{CRF}} = -\log P_{\text{CRF}}$ and P_{CRF} is defined in Eq. (1). For the sequence generator, at the first 10k steps, we train it with $\mathcal{L}_{\text{LM}}$ as described in Eq. (2). Then, we incorporate the structure-aware RL objective (Eq. (3)) and further train the sequence generator with $\mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{RL}}$ for 5k more steps.

5 Experiment Setup

5.1 Datasets and Evaluation Metrics

ToTTo Dataset (Parikh et al., 2020) consists of Wikipedia tables paired with human-written descriptions. Each input is a full table with highlighted cells and the model is required to generate the text that describes the highlighted cells. Similar to previous studies (Parikh et al., 2020; Kale and Rastogi, 2020), we only use the highlighted cells

[2] https://github.com/google-research-datasets/ToTTo

Model	Overall			Overlap			Non-Overlap		
	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT
NCP	19.2	29.2	-0.576	24.5	32.5	-0.491	13.9	25.8	-0.662
Pointer-Generator	41.6	51.6	0.076	50.6	58.0	0.244	32.2	45.2	-0.092
BERT-to-BERT	44.0	52.6	0.121	52.7	58.4	0.259	35.1	46.8	-0.017
T5-3B	**49.5**	58.4	0.230	**57.5**	62.6	0.351	41.4	54.2	0.108
Ours	49.2	**58.7**	**0.249**	56.9	**62.8**	**0.371**	**41.5**	**54.6**	**0.126**

Table 2: ToTTo test set results: All reported results, including ours, can be found in the official Leaderboard.[2]

as the model input. We report the automatic result of BLEU-4, PARENT[3] (Dhingra et al., 2019), and a learnt metric BLEURT (Sellam et al., 2020). Note that ToTTo features a hidden test set with two splits: Overlap and Non-Overlap. The Non-Overlap set contains out-of-domain examples. To get the test set result, a submission must be made to the leaderboard.

WebNLG Dataset is used in the WebNLG challenge (Gardent et al., 2017). For each data instance, the input is a set of RDF triples from DBPedia and the output is their textual description. The test set of WebNLG features a Seen and Unseen subset. The Unseen subset contains out-of-domain instances. Following previous studies, we report the the automatic result of BLEU and METEOR (Banerjee and Lavie, 2005).

5.2 Implementation Details

Our implementation is based on the Huggingface Library (Wolf et al., 2019). We optimize the model using Adam (Kingma and Ba, 2015) with a learning rate of 2e$-$5 and a batch size of 64.

6 Results

In this section, we report the experimental results.

6.1 ToTTo Results

We compare our model with the latest models on ToTTo dataset, including NCP (Puduppully et al., 2019a), Pointer-Generator (See et al., 2017), BERT-to-BERT (Rothe et al., 2020) and T5-3B (Kale and Rastogi, 2020). Similar to our model, the later two are also based on pre-trained language models.

Table 2 lists the results on ToTTo test set. For most of the metrics, our model with 140M parameters outperforms the current state-of-the-art T5-3B model which has over 2.8B parameters. The results on the PARENT metric suggest that our model can generate more factually accurate text. Moreover, in the Non-Overlap subset, our model achieves the best result on all metrics, showing its robustness to out-of-domain examples.

6.2 WebNLG Results

We compare our approach with two types of models on WebNLG dataset. The first type of models does not use pre-trained language models (PLMs), including GTR-LSTM (Trisedya et al., 2018), Transformer (Ferreira et al., 2019), Step-by-Step (Moryossef et al., 2019), and PLANENC (Zhao et al., 2020). Similar to ours, the latter three are pipeline models that utilize different methods to decide the output planning before generating the result. The second line of research utilizes PLMs, including Switch-GPT (Chen et al., 2020b), T5 (Kale and Rastogi, 2020), and T5+Prefix (Ribeiro et al., 2020). The Switch-GPT model applies a copy mechanism to copy content from the source to the output. We also include the top systems of the WebNLG challenge, including ADAPT, TILB-SMT, and MELBOURNE.

Evaluation on Text Generation. Table 3 lists the results of different methods in terms of text generation. We see that our approach outperforms all prior works. Compared with previous models that utilize PLMs, our performance improvements suggest that the incorporation of an explicit content plan can provide effective guiding signal for the model to achieve better generation results.

Evaluation on Content Planning. Next, we compare our content planner with other pipeline models in terms of content planning performance. Following Zhao et al. (2020), we report the results on planning accuracy (P-A) and planning BLEU-2 score (B-2) against the human-generated plans[4]. In addition, we examine two ablated variants of our

[3]PARENT is a word-overlap based metric that reflects the factual accuracy of the generated text in relation to both the input table and the reference sentence.

[4]The human-generated plans are provided in the enriched WebNLG dataset (Ferreira et al., 2018).

Model	Seen		Unseen		Overall	
	B.	M.	B.	M.	B.	M.
ADAPT†	60.59	0.44	10.53	0.19	31.06	0.31
TILB-SMT†	54.29	0.42	29.88	0.33	44.28	0.38
MELBOURNE†	54.52	0.41	33.27	0.33	45.13	0.37
GTR-LSTM†	54.00	0.37	29.20	0.28	37.10	0.31
Transformer†	56.28	0.42	23.04	0.21	47.24	0.39
Step-by-Step†	53.30	0.44	38.23	0.34	47.24	0.39
PLANENC†	64.42	0.45	38.23	0.37	52.78	0.41
Based on PLMs						
Switch-GPT	60.98	0.43	40.67	0.34	52.17	0.40
T5‡	63.90	0.46	52.80	0.41	57.10	0.44
T5+Prefix‡	64.71	0.45	53.67	0.42	59.70	0.44
Ours	**65.42**	**0.48**	**54.52**	**0.44**	**60.51**	**0.46**

Table 3: Text generation results on WebNLG datasets, where B. and M. represent BLEU and METEOR metrics. † and ‡ results are cited from Zhao et al. (2020) and Ribeiro et al. (2020), respectively.

Model	Seen		Unseen		Overall	
	Acc.	B-2	Acc.	B-2	Acc.	B-2
Transformer†	0.56	74.30	0.09	20.90	0.34	49.30
GRU†	0.56	75.80	0.10	25.40	0.35	52.20
Step-by-Step†	0.49	73.20	0.44	68.00	0.47	70.80
PLANENC†	0.63	80.80	0.61	79.30	0.62	80.10
Ours	**0.74**	**86.01**	**0.70**	**83.79**	**0.72**	**84.97**
w/o CRF	0.67	82.92	0.63	80.65	0.65	81.73
w/o PLMs	0.70	84.05	0.65	81.98	0.68	83.02

Table 4: Evaluation results on content planning. † results are copied from Zhao et al. (2020).

content planner by either removing the CRF layer (w/o CRF) or using randomly initialized parameters instead of the pre-trained BERT (w/o PLMs). Table 4 lists the results. We see that our content planner outperforms all the baselines on both measures. Moreover, the results show that both the CRF layer and the pre-trained parameters positively contribute to the overall performance which further justifies our design of the content planner.

6.3 Human Evaluation

We also conduct a human evaluation to assess our model, using graders proficient in English from an internal grading platform. We randomly selected 200 samples from the ToTTo validation set. For each sample, we first use our sequence generator to produce the result with the content plan (CP) predicted by the content planner. Next, we randomly shuffle the predicted content plan and generate five different results (Shuffled CP). For comparison, we also include results of BERT-to-BERT and T5-3B using greedy decoding. All generated results, plus the reference sentence, are evaluated by three graders on a 3-point Likert scale (0, 1, or 2) for each of the following features[5]:

	Faithfulness	Fluency	Accuracy
Agreement	0.663	0.617	0.518
Reference	1.819	1.762	1.753
BERT-to-BERT	1.589	1.593	-
T5-3B	1.701	1.696	-
Ours(CP)	1.794	1.753	1.742
Ours(Shuffled CP)	1.778	1.746	1.552

Table 5: Human Evaluation Results

- **Faithfulness**: Whether the sentence is factually consistent with the input data.
- **Fluency**: Whether the sentence is fluent and easy to understand.
- **Accuracy**: How accurately the sentence follows the input content plans[6].

Table 5 lists the results, with the first row showing strong inter-annotator agreements as measured by Fleiss′ kappa coefficient (Fleiss et al., 1971). Comparing with BERT-to-BERT and T5-3B, our model achieves best results on both measures. Furthermore, on the faithfulness and fluency metrics, our model with both CP and Shuffled CP performs comparably with the reference sentence (Sign Test with p-value > 0.4). On the accuracy metric, our CP model also performs comparably with the reference as judged by the Sign Test. However, with randomly shuffled content plan, our model (Shuffled CP) fails to match the accuracy of the reference (p-value < 0.05). Our analysis is that the random content plans could contain patterns that are rare or unseen during training. In such cases, our model might fail to produce results that precisely follow the content plan, resulting in a lower accuracy score. Nonetheless, the human results suggest that, while being able to produce fluent and correct sentences, our model is also highly controllable. Finally, we note that on the accuracy metric, even the reference sentence does not score a perfect 2.0. This suggests that our simple heuristic delexicalizer $\mathcal{F}$ introduced in §3 still lapses behind human performance. We leave to future work of designing better $\mathcal{F}$.

7 Further Analysis

In this section, we present and discuss more empirical analyses of the proposed model.

[5]More evaluation details are provided in the Appendix A.

[6]As BERT-to-BERT and T5-3B do not take the content plan as input, thus we do not report their accuracy score.

Model		Quality		Diversity	
		BLEU	PARENT	Self-BLEU↓	iBLEU
B2B	Greedy	44.15	53.08	100.00	15.32
	Beam	41.58	49.87	75.04	18.26
	Top-k	42.47	50.43	82.20	17.54
	Nucleus	42.92	50.91	84.26	17.48
T5-3B	Greedy	48.43	57.80	100.00	18.74
	Beam	45.12	55.20	83.68	19.36
	Top-k	46.31	55.90	88.86	19.28
	Nucleus	46.53	56.30	90.11	19.20
Ours					
Predict	CP	**49.10**	**58.27**	100.00	19.28
	Shuffled CP	40.75	51.96	**25.91**	**27.42**
Oracle	CP	54.43	62.75	100.00	23.54
	Shuffled CP	42.99	56.17	26.90	29.01

Table 6: Experimental results on the overall ToTTo validation set, where ↓ means lower is better.

7.1 Evaluation on Generation Diversity

Setup. We first evaluate the ability of different models in generating diverse results on the overall ToTTo validation set. We compare our model with two strong baselines, BERT-to-BERT (B2B) and T5-3B. Given the input data, the baseline models generate the results with different decoding strategies[7], including greedy search, beam search (beam size of 10), top-k sampling ($k = 50$) (Fan et al., 2018), and Nucleus sampling ($p = 0.9$) (Holtzman et al., 2020). For our model, to generate diverse results, we simply vary the input content plan and use greedy decoding. We use two variants of the input content plan: (1) the content plan predicted by the content planner (Predict), or (2) the reference content plan (Oracle). For each variant, five results are generated by either using the input content plan (CP), or using five randomly shuffled forms of the content plan (Shuffled CP). The outputs are expected to vary in the latter case only.

Metric. To measure the output quality, BLEU and PARENT scores are reported. To evaluate the generation diversity, we use Self-BLEU (Zhu et al., 2018) and iBLEU (Sun and Zhou, 2012) metrics[8].

Results. Table 6 lists the results in which our model ranks best on all metrics. On the quality metrics, we observe notable performance improvements from our model by using the reference content plan (Oracle), suggesting that the choice of content plan has a significant impact on the outputs. By shuffling the content plan, our model shows the largest decrease in BLEU and PARENT, showing that the variation of content plan encourages our model to produce diverse results that have different structures than the reference.

Furthermore, we see that, even with different decoding strategies, the baseline models still generate results that are very similar to the ones acquired from greedy search, with their BLEU and PARENT scores relatively unchanged. The results on the diversity metrics also verify the superiority of our model which outperforms the strong T5-3B model by over 57 and 8 points on Self-BLEU and iBLEU[9]. The performance gains suggest that the controllable property of our model is beneficial in producing high-quality as well as diverse results.

Model	CP	RL	Type	BLEU	PARENT	S-BLEU
1	×	×	-	47.50	56.92	43.87
2	×	✓	-	48.10	57.34	48.93
3	✓	×	Predict	48.53	57.87	57.92
			Oracle	53.82	61.99	75.59
Ours	✓	✓	Predict	**49.10**	**58.27**	**62.27**
			Oracle	54.43	62.75	80.32

Table 7: Ablation Studies on the overall ToTTo validation set. Model 1 gives a baseline for the BART model.

7.2 Ablation Study

In this part, we evaluate the importance of each component of our model on the overall ToTTo validation set. Specifically, we study the effect of content plan (CP) and the RL training by removing them iteratively. In addition to BLEU and PARENT, we measure the structure of the model output against the reference content plan with a S-BLEU metric. Given the data T, the reference content plan C, and the model output S', S-BLEU is defined as $B(C, C')$, where $B(\cdot, \cdot)$ measures the BLEU score, $C' = \mathcal{F}(T, S')$, and $\mathcal{F}$ is the heuristic delexicalizer described in §3. The results are listed in Table 7 with the first row showing the baseline results of BART model.

Necessity of Content Plan. By comparing models with and without the content plan (model 1 vs. 3 and model 2 vs. ours), we observe that the content plan is an effective guiding signal that leads to better results. Moreover, we see that the Oracle results outperform the Predict results by a large margin, showing that the quality of the content plan is an important factor of the model performance and future research can focus more on this aspect.

[7]For each decoding strategy, five results are generated.

[8]For all evaluation metrics, we use the same hyper-parameters as in the original works that proposed the metric.

[9]By definition, models using greedy search get 100 Self-BLEU as the generated results are always the same.

Table: **Title**[George Washington Colonials football] **Date**[1956] **Game**[Sun Bowl] **Result**[W 13-0] **Opponent**[Texas Western] **Notes**[Bowl Games]	
Reference: In 1956, George Washington Colonials scored 13–0 against Texas Western at the Sun Bowl.	
T5-3B: *Greedy Search*	**Ours:** *CP*
George Washington Colonials football won the Sun Bowl (1956) over Texas Western.	**ICP**: Date → Title → Result → Opponent → Game In 1956, George Washington Colonials football team scored 13–0 against Texas Western in the Sun Bowl.
T5-3B: *Beam Search*	**Ours:** *Shuffled CP*
1: George Washington Colonials football won the 1956 Sun Bowl against Texas Western.	**ICP**: Date → Result → Opponent → Title → Game In 1956, with a 13–0 victory over Texas Western, the Colonials football team won the Sun Bowl.
2: George Washington Colonials won the 1956 Sun Bowl against Texas Western.	**ICP**: Title → Game → Date → Result → Opponent George Washington Colonials football won the Sun Bowl in 1956 with a 13–0 victory over Texas Western.
3: In 1956, George Washington Colonials won the Sun Bowl against Texas Western.	**ICP**: Game → Result → Title → Opponent → Date In the Sun Bowl, a 13–0 victory for George Washington Colonials over Texas Western in 1956.
4: George Washington Colonials won the Sun Bowl against Texas Western in 1956.	**ICP**: Title → Opponent → Game → Result → Date George Washington Colonials football team defeated Texas Western in the Sun Bowl, with 13–0, in 1956.
5: George Washington Colonials football won the 1956 Sun Bowl over Texas Western.	**ICP**: Opponent → Game → Date → Result → Title The Colonials defeated Texas Western in the Sun Bowl 1956, with a 13–0 score, by George Washington Colonials.

Table 8: Case study on ToTTo dataset. Given the input data, we present the generated results from various models using different decoding strategies. **ICP** denotes the "input content plan". (Best viewed in color)

Tripleset	(Alan Bean | nationality | United States), (Alan Bean | occupation | Test pilot), (Alan Bean | birthPlace | Wheeler , Texas), (Alan Bean | selectedByNASA | 1963), (Alan Bean | status | "Retired")
Reference	Alan Bean is a US national born in Wheeler, Texas. He is a retired test pilot who joined NASA in 1963.
Ours (Shuffled CP)	**ICP**: nationality → birthPlace → selectedByNASA → status → occupation Alan Bean is a US national who was born in Wheeler, Texas. He was selected by NASA in 1963 and is now retired. He was a test pilot.
	ICP: nationality → occupation → selectedByNASA → birthPlace → status Alan Bean is a US national who served as a test pilot and was selected by NASA in 1963. He was born in Wheeler, Texas and is now retired.
	ICP: selectedByNASA → occupation → status → birthPlace → nationality Alan Bean was selected by NASA in 1963 as a test pilot. He is now retired. He was born in Wheeler, Texas and is a United States national.

Table 9: Case study of our model's results on WebNLG dataset. (best viewed in color)

Effect of RL. By comparing the models trained with and without RL (model 1 vs. 2 and model 3 vs. ours), we see that training with our proposed RL objective consistently improves the model performance. The most notable improvement is observed in S-BLEU which means that the generated outputs better follow the input content plan. This is in line with our hypothesis that our reward function in Eq. (4) helps to improve the model's adherence to the output structure defined by the content plan.

7.3 Case Study

To gain more insights into our model, we present generated examples from ToTTo and WebNLG datasets[10] in Table 8 and Table 9, respectively.

Quality. In Table 8, we compare our model with predicted content plan against T5-3B. We see that T5-3B fails to produce the key game result (i.e. *13-0*) in its outputs. In contrast, by following the content plan, our model is able to maintain all key information in its generated results.

Diversity and Controllability. Next, we examine the output diversity and controllability. For the T5-3B model, when using beam search, only the position of the term *"1956"* varies, showing its reduced ability to generate diverse outputs. For our model, the variation of content plan leads to outputs with diverse structures. Furthermore, the results show that our model is not only able to control the intra-sentence output structure as shown in Table 8 but also to control the inter-sentence output structure as shown in Table 9.

Error Analysis. We show one failure case in the bottom right cell of Table 8, in which it repeats the *Title* key twice in the output. Our analysis for such error is that the randomly shuffled content plan might contain patterns that are rarely seen in training. One possible solution is filtering out rare content plan patterns via statistical approaches such as bigram statistics.

8 Conclusion

In this study, we propose a new Plan-then-Generate (PlanGen) framework for data-to-text generation

[10]More examples are shown in the Appendix B.

which can be easily applied to data with different structures. Extensive experiments and analyses are conducted on two benchmark datasets. Both automatic and human evaluation results demonstrate that our model is highly controllable. Furthermore, compared with previous studies, our model achieves better results both in terms of the generation quality as well as the output diversity. Our code, models and other related resources can be found in https://github.com/yxuansu/PlanGen/

Acknowledgments

The authors wish to thank Ehsan Shareghi, Zaiqiao Meng, Piji Li, and Benjamin Muller for their insightful discussions and support. Many thanks to our anonymous reviewers for their suggestions and comments.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. Open question answering over tables and text. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020a. KGPT: knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8635–8648. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020b. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 183–190. Association for Computational Linguistics.

Émilie Colin and Claire Gardent. 2019. Generating text from anonymised structures. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 112–117. Association for Computational Linguistics.

Émilie Colin, Claire Gardent, Yassine Mrabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. The webnlg challenge: Generating text from dbpedia data. In *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK*, pages 163–167. The Association for Computer Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4884–4895. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. 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, Tilburg University, The Netherlands, November 5-8, 2018*, pages 171–176. Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 552–562. Association for Computational Linguistics.

J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. 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. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

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. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 97–102. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical NLG framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1406–1415. The Association for Computer Linguistics.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, June 15-17, 1983*, pages 145–150. ACL.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Ziran Li, Zibo Lin, Ning Ding, Hai-Tao Zheng, and Ying Shen. 2020. Triple-to-text generation with an anchor-to-prototype framework. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3780–3786. ijcai.org.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2047–2057. Association for Computational Linguistics.

Kathleen R. McKeown. 1992. *Text generation - using discourse strategies and focus constraints to generate natural language text*. Studies in natural language processing. Cambridge University Press.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2267–2277. 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*, pages 201–206. Association for Computational Linguistics.

Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.

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. ACL.

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1173–1186. Association for Computational Linguistics.

Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: A parametric theory and its instantiations. *Comput. Linguistics*, 30(3):309–363.

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

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text generation with entity modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2023–2035. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *CoRR*, abs/2007.08426.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Trans. Assoc. Comput. Linguistics*, 8:264–280.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentations in spoken dialog systems. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 79–86, Barcelona, Spain.

Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. 2021a. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 234–243. Association for Computational Linguistics.

Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021b. Few-shot table-to-text generation with prototype memory. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 38–42. The Association for Computer Linguistics.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1627–1637. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve J. Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 120–129. The Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3174–3187. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

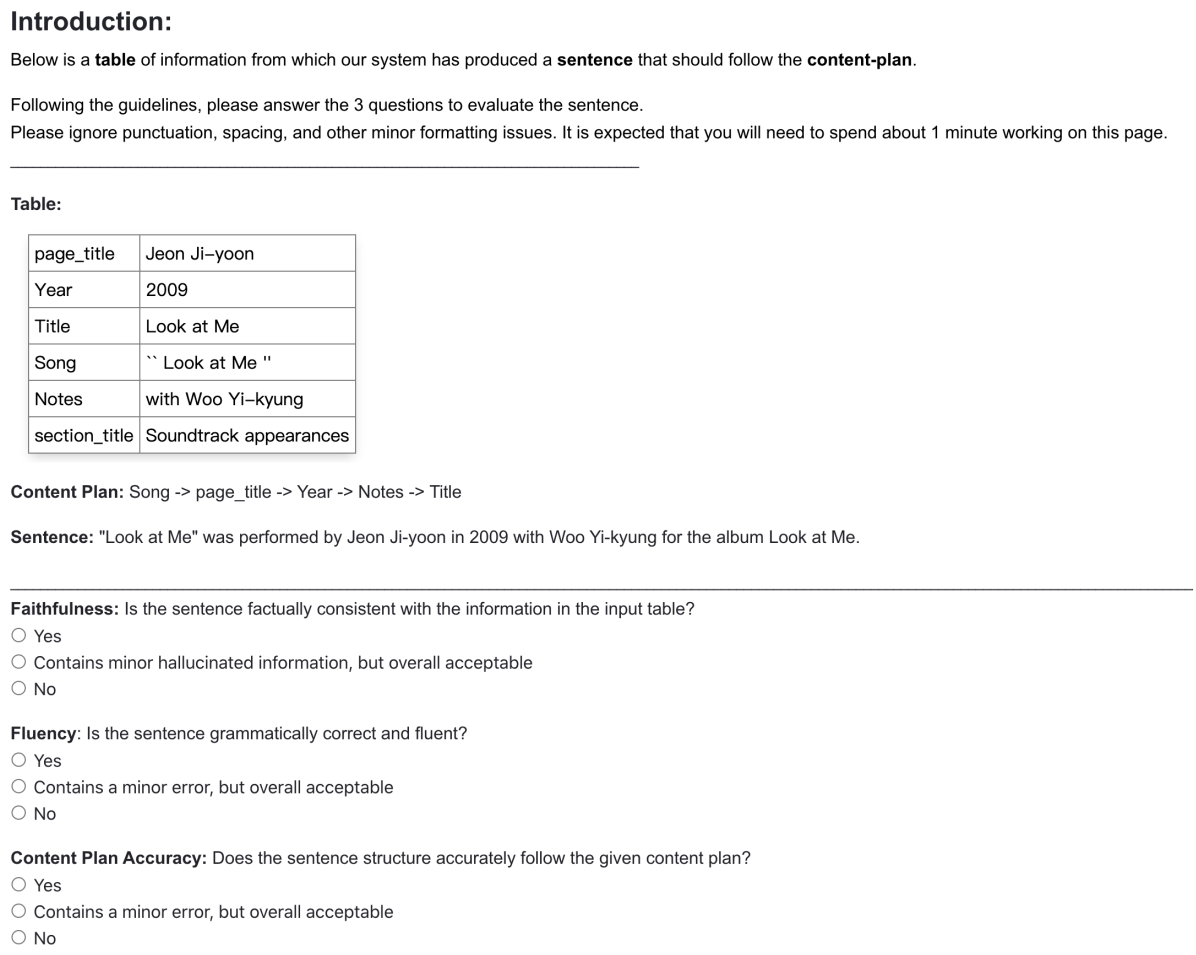
Rong Ye, Wenxian Shi, Hao Zhou, Zhongyu Wei, and Lei Li. 2020. Variational template machine for data-to-text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chao Zhao, Marilyn A. Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2481–2491. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

A Details of Human Evaluation Setup

To perform human evaluation, we randomly sample 200 samples from the ToTTo validation set. For each sampled data, we use each baseline model (BERT-to-BERT and T5-3B) to produce one result. As for our model, we produce 6 different results (one with the predicted content plan, the other five with five randomly shuffled versions of the predicted content plan). Therefore, for each case, we have 9 different results (1 from BERT-to-BERT, 1 from T5-3B, 6 from our model, and 1 reference). To reduce human bias, we randomly shuffle these 1800 data points before presenting them to three annotators. Each annotator is asked to assess all these 1800 data points. Because BERT-to-BERT and T5-3B do not take the content plan as input, thus we only measure the accuracy score for the results generated by our model and the reference sentence. Note that the accuracy score of the reference sentence is measured against the reference content plan. In Figure 3, we show an example of the human evaluation interface.



Introduction:

Below is a **table** of information from which our system has produced a **sentence** that should follow the **content-plan**.

Following the guidelines, please answer the 3 questions to evaluate the sentence.
Please ignore punctuation, spacing, and other minor formatting issues. It is expected that you will need to spend about 1 minute working on this page.

Table:

page_title	Jeon Ji–yoon
Year	2009
Title	Look at Me
Song	`` Look at Me ''
Notes	with Woo Yi–kyung
section_title	Soundtrack appearances

Content Plan: Song -> page_title -> Year -> Notes -> Title

Sentence: "Look at Me" was performed by Jeon Ji-yoon in 2009 with Woo Yi-kyung for the album Look at Me.

Faithfulness: Is the sentence factually consistent with the information in the input table?
○ Yes
○ Contains minor hallucinated information, but overall acceptable
○ No

Fluency: Is the sentence grammatically correct and fluent?
○ Yes
○ Contains a minor error, but overall acceptable
○ No

Content Plan Accuracy: Does the sentence structure accurately follow the given content plan?
○ Yes
○ Contains a minor error, but overall acceptable
○ No

Figure 3: Example of Human Evaluation Interface

B More Examples of Generated Result

In this part, we provide more generated examples of our model. The generated results on samples from WebNLG and ToTTo datasets are shown in Table 10 and 11, respectively. From the results, we can see that our model is able to generate fluent and diverse sentence while maintaining the structure defined by the desired content plan. In particular, our model is able to control the output structure both on the inter-sentence level (i.e. the structure across multiple sentences) as shown in Table 9 and on the intra-sentence level (i.e. the structure within a single sentence) as shown in Table 11. These results further demonstrate the applicability and generalization ability of our model.

Tripleset	(Allama Iqbal International Airport | location | Pakistan), (Allama Iqbal International Airport | runwayLength | 2900.0), (Allama Iqbal International Airport | cityServed | Lahore), (Allama Iqbal International Airport | operatingOrganisation | Pakistan Civil Aviation Authority)
Reference	Allama Iqbal International Airport is located in Lahore at Pakistan. It has a runway length of 2900 and is operated by the Pakistan Civil Aviation Authority.
Ours (Shuffled CP)	**ICP**: cityServed → location → runwayLength → operatingOrganisation Allama Iqbal International Airport serves the city of Lahore and is located in Pakistan. The runway length is 2900.0 and the airport is operated by the Pakistan Civil Aviation Authority.
	ICP: operatingOrganisation → cityServed → location → runwayLength The Pakistan Civil Aviation Authority is the operating organisation of the Allama Iqbal International Airport which serves the city of Lahore in Pakistan. The airport has a runway length of 2900.0.
	ICP: runwayLength → cityServed → location → operatingOrganisation Allama Iqbal International Airport has a runway length of 2900.0 and serves the city of Lahore in Pakistan. The airport is operated by the Pakistan Civil Aviation Authority.
	ICP: location → operatingOrganisation → cityServed → runwayLength Allama Iqbal International Airport is located in Pakistan and is operated by the Pakistan Civil Aviation Authority. The airport serves the city of Lahore and has a runway length of 2900.0.
	ICP: runwayLength → operatingOrganisation → cityServed → location Allama Iqbal International Airport has a runway length of 2900.0 and is operated by the Pakistan Civil Aviation Authority. The airport serves the city of Lahore and is located in Pakistan.
Tripleset	(Turkey | leaderTitle | president of Turkey), (Turkey | leader | Ahmet Davutoglu), (Turkey | capital | Ankara), (Turkey | largestCity | Istanbul), (Atatürk Monument (Izmir) | location | Turkey)
Reference	The Ataturk Monument (Izmir) is located in Turkey, whose largest city is Istanbul and whose capital is Ankara. President of Turkey is the official title of the Turkish leader and his name is Ahmet Davutoglu.
Ours (Shuffled CP)	**ICP**: location → leaderTitle → capital → leader → largestCity The Atatürk Monument (Izmir) is located in Turkey, where the leader is known as the president and the capital city is Ankara. Ahmet Davutoglu is the leader of Turkey and the largest city is Istanbul.
	ICP: capital → leaderTitle → leader → largestCity → location Ankara is the capital of Turkey, which is lead by president Ahmet Davutoglu. The largest city is Istanbul and the Atatürk Monument is located in Turkey.
	ICP: largestCity → location → capital → leaderTitle → leader Istanbul is the largest city in Turkey, where the Ataturk Monument is located and the capital is Ankara. The leader of Turkey is known as the president and Ahmet Davutoglu is the leader.
	ICP: location → capital → leader → largestCity → leaderTitle The Atatürk Monument (Izmir) is located in Turkey, where the capital is Ankara, the leader is Ahmet Davutoglu and the largest city is Istanbul. The leader of Turkey is known as the president of Turkey.
	ICP: location → largestCity → capital → leaderTitle → leader The Atatürk Monument (Izmir) is located in Turkey, where the largest city is Istanbul and the capital is Ankara. The leader of Turkey is known as the president and Ahmet Davutoglu is the leader.

Table 10: Examples of generated result from WebNLG dataset, where **ICP** denotes "input content plan". The expressions correspond to different contents are highlighted with different colors. (best viewed in color)

Input Table
Page_Title[List of New Zealand records in swimming] **Event**[400m freestyle] **Time**[4:03.63] **Name**[Lauren Boyle] **Club**[New Zealand] **Date**[29 July 2012] **Meet**[Olympic Games] **Location**[London, United Kingdom] **Section_Title**[Women]
Reference Sentence
At the 2012 Olympics in London, Boyle was fourth fastest in the heats of the 400m freestyle in a New Zealand record 4:03.63.
Controlled Data-to-Text Generation
ICP: Date → Name → Event → Time → Location → Meet → Page_Title On 29 July 2012, New Zealand's Lauren Boyle finished the 400 metre freestyle in 4:03.63 at London Olympics, which was a New Zealand record.
ICP: Page_Title → Location → Meet → Name → Event → Time → Date The New Zealand swimming record was set in London at the 2012 London Olympics, where Lauren Boyle finished the 400 metre freestyle in 4:03.63, in July 2012
ICP: Meet → Location → Date → Name → Time → Event → Page_Title At the Olympic Games in London in July 2012, Lauren Boyle finished in 4:03.63 in the 400 metre freestyle, setting a New Zealand record.
ICP: Name → Event → Date → Meet → Location → Page_Title → Time Lauren Boyle competed in the 400 metre freestyle at the 2012 Olympic Games in London, finishing with a New Zealand record time of 4:03.63.
ICP: Location → Event → Name → Time → Page_Title → Date → Meet In London, in the 400m freestyle, Lauren Boyle finished in 4:03.63, setting a New Zealand record at the 2012 Olympic Games.
Input Table
Page_Title[Kazunari Murakami] **Result**[Loss] **Opponent**[Akihiro Gono] **Method**[KO (head kick)] **Date**[October 13, 1995] **Event**[Lumax Cup: Tournament of J '95] **Section_Title**[Mixed martial arts record]
Reference Sentence
Murakami began his career fighting in the Lumax Cup: Tournament of J '95 on October 13, 1995, where he lost his first fight to Akihiro Gono by knockout.
Controlled Data-to-Text Generation
ICP: Page_Title → Date → Event → Opponent Kazunari Murakami made his debut on October 13, 1995 at Lumax Cup: Tournament of J '95, losing to Akihiro Gono by KO.
ICP: Event → Date → Page_Title → Opponent At Lumax Cup: Tournament of J '95 on October 13, 1995, Kazunari Murakami lost to Akihiro Gono by KO.
ICP: Opponent → Page_Tilte → Event → Date Akihiro Gono defeated Kazunari Murakami at Lumax Cup: Tournament of J '95 on October 13, 1995.
ICP: Date → Opponent → Page_Title → Event On October 13, 1995, Akihiro Gono defeated Kazunari Murakami at Lumax Cup: Tournament of J '95.
ICP: Event → Page_Title → Opponent → Date At Lumax Cup: Tournament of J '95, Kazunari Murakami lost to Akihiro Gono by KO on October 13, 1995.
Input Table
Page_Title[Reform Party of the United States of America] **Year**[2008] **Name**[Frank McEnulty] **Home_state**[California] **Section_Title**[Presidential tickets]
Reference Sentence
Frank McEnulty of California, was nominated to be the Reform Party's 2008 presidential candidate.
Controlled Data-to-Text Generation
ICP: Year → Page_Title → Name → Home_state → Section_Title In 2008, the Reform Party of the United States of America nominated Frank McEnulty of California as its presidential candidate.
ICP: Page_Title → Section_Title → Home_state → Year → Name Reform Party of the United States of America nominated its first presidential nominee from California in 2008, Frank McEnulty.
ICP: Home_state → Name → Section_Title → Year → Page_Title California's Frank McEnulty was nominated as presidential candidate in 2008 by the Reform Party of the United States of America.
ICP: Page_Title → Name → Home_state → Section_Title → Year Reform Party of the United States of America nominated Frank McEnulty of California as its presidential candidate in 2008.
ICP: Year → Name → Page_Title → Home_state → Section_Title In 2008, Frank McEnulty of Reform Party of the United States of America from California ran for the presidential election.

Table 11: Examples of generated result from ToTTo dataset, where **ICP** denotes "input content plan". The expressions correspond to different contents are highlighted with different colors. (best viewed in color)