Improving User Controlled Table-To-Text Generation Robustness

Hanxu Hu[1] , Yunqing Liu[2], Zhongyi Yu[1] and Laura Perez-Beltrachini [1]
[1] School of Informatics, University of Edinburgh, United Kingdom
[2] The Hong Kong Polytechnic University, HongKong
{huhanxu1233,lyq6175215241,zhongyics}@gmail.com
lperez@exseed.ed.ac.uk

Abstract

In this work we study user controlled table-to-text generation where users explore the content in a table by selecting cells and reading a natural language description thereof automatically produce by a natural language generator. Such generation models usually learn from carefully selected cell combinations (clean cell selections); however, in practice users may select unexpected, redundant, or incoherent cell combinations (noisy cell selections). In experiments, we find that models perform well on test sets coming from the same distribution as the train data but their performance drops when evaluated on realistic noisy user inputs. We propose a fine-tuning regime with additional user-simulated noisy cell selections. Models fine-tuned with the proposed regime gain 4.85 BLEU points on user noisy test cases and 1.4 on clean test cases; and achieve comparable state-of-the-art performance on the ToTTo dataset.[1]

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

RUSHING							RECEIVING				
YEAR	**TEAM**	**ATT**	**YDS**	**AVG**	**LNG**	**TD**	**NO.**	**YDS**	**AVG**	**LNG**	**TD**
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Target Text: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Figure 1: An example in the ToTTo dataset. The figure is retrieved from (Parikh et al., 2020). The cells coloured in yellow are the highlight cells.

1 Introduction

The goal of table-to-text generation is to provide the user with a description of the most relevant content in a given table (Lebret et al., 2016; Wiseman et al., 2018; Perez-Beltrachini and Lapata, 2018; Puduppully et al., 2019). Recently, Parikh et al. (2020) proposed a controlled table-to-text generation task where the goal is to automatically create a description for a determined subset of the table, namely the highlighted table cells. The main focus on Parikh et al.'s 2020 work is to assess the performance of neural text generators in a more controlled setting, i.e., when given an input table with explicit instructions (i.e., highlights) on what should be expressed in the output description. In this work, we view this task in the context of a natural language interface, as a *user controlled table-to-text* generation task, where users provide those highlights interactively by exploring the content of a given table and study these user interactions. Figure 1 illustrates the case where a user selects some cells (highlighted in yellow) and the generator provides a description thereof (shown below the table).

A crucial aspect of usability assessment for a generator in this interactive table-to-text task is robustness. In a recent study by (Mille et al., 2021), it has been shown that neural generation models fail to maintain their in distribution performance when confronted with realistic scenarios at test time such as typos in the input text. In the case of user controlled table-to-text generation, users may introduce noise when exploring the table content and select cell combinations that turn out to be unexpected, redundant, or incoherent. For example, in Figure 1, when the user wants to express "eleven seasons", they might miss one year or highlight the header cell. They may also select unrelated headers, for instance adding the header "LNG" to the current selection. Existing controlled table-to-text generation models (Parikh et al., 2020; Su et al., 2021; Kale and Rastogi, 2020) are trained on carefully selected cell combinations (**clean** cell highlights) from the ToTTo dataset (Parikh et al., 2020). We argue that these models will not generalize well in practice with user **noisy** highlights. No previous work has study model robustness under this practical set up.

[1]Our code is available at
https://github.com/hanxuhu/controllT2Trobust

Findings of the Association for Computational Linguistics: EACL 2023, pages 2317–2324
May 2-6, 2023 ©2023 Association for Computational Linguistics

We carry out a usability study to observe how users highlight cells in a table. Based on the imperfect cell selections that users produce, we automatically create additional data examples by corrupting examples from the original ToTTo dataset. We then fine-tune state-of-the-art table-to-text neural generation models with this additional data. We compare the performance of models fine-tuned only with clean cell highlights versus those trained with additional noisy cell highlights, both on a test set with clean and noisy highlights. Experimental results show that models fine-tuned with clean cell highlights only perform well on clean test cases (i.e., performance drops dramatically when evaluated on noisy cell highlights). That is, these models do not generalise well in practice with user noisy cell selections. In contrast, the proposed training scheme with additional noisy cell highlights not only makes user controlled table-to-text models achieve better performance in practical scenarios, but it also boosts performance on perfect inputs. Experimental results show that models fine-tuned with our proposed training regime gain 4.85 BLEU points on noisy and 1.4 BLEU points on clean highlights; and achieve comparable state-of-the-art performance on the ToTTo dataset.[2]

2 Methodology

We describe the process for creating user noisy cell highlights from examples in ToTTo (Parikh et al., 2020) (§2.1 and §2.2). Then, we evaluate models optimized with the standard training scheme (i.e., only on clean cell highlights) on the created noisy test cases. Results show that these models perform poorly. To improve model robustness, we propose a new learning regime described in §2.3. To further improve performance, we fine-tune with Reinforcement Learning (RL) based optimisation (§2.4). Finally, §2.5 summarises the learning schemes and objective functions we propose for robust user controlled table-to-text generation.

2.1 How Do Users Select Cells?

To understand how users proceed when exploring a table and selecting cells we carry out a human study using examples from the ToTTo dataset. Participants are given a plain table (i.e., without highlights) and asked to highlight cells according to an exploratory intention. For a more controlled setting, we give the sentence associated to the table as the exploratory intention. In this way, we avoid ambiguous post-selection analysis of what the user intention was. In addition, this allows us to compare user selections with reference highlights as well as differences (if any) in model generated texts given user and reference highlights.

We conduct this study on Amazon Mechanical Turk (the interface is described in Appendix C). We collect 90 user highlights (3 participants, volunteers known by the authors, and 30 examples from the validation set) and observe the following noise in their highlights. Participants apply different criteria to include (or not) table headers; select additional cells in columns/rows around cells containing relevant content; and do not select cells that contain content relevant to the intention.

2.2 Creating User Noisy Cell Selections

Given the input table T, the reference text S, and the reference highlight cells $H \in T$ relevant for generating S, we create noisy user cell selections as follows. We provide an example illustrating each noise type in Figure 2.

Noise 1: Additional Table Cells In practical scenarios, users may accidentally select random cells that are not related to their exploration intention. Thus, we randomly select k cells from the table cells in T that are not in H and add them into H to form a corrupted input H_1. H_1 can be viewed as adding irrelevant information in the generation of the target text S.

Noise 2: Table Headers as Additional Inputs Reference highlight cells in the ToTTo dataset do not cover table headers. As we have observed, users may decide to include (or not) table headers in different cases. To simulate this, we first retrieve table headers corresponding to highlight cells in H. Then, we randomly select k unique headers and add them into H to get the corrupted input H_2.

Noise 3: Similar Table Cells For this type of noise, we select cells that are in the same row/column as the highlight cells. The intuition, as seen in the user study, is that these cells will have similar semantics to those cells underlying the exploratory intention and users tend to select them. For H_3, we first retrieve table cells that are in the same row/column as highlight cells. Then, we randomly select k unique cells thereof and add them into H.

[2] ToTTo leaderboard.

Noise 4: Remove Cells from H Users also miss some of the highlight cells in H. For this type of noise, we first retrieve those cells in H that are irrelevant (i.e., their content is not expressed in) for generating S. After getting the irrelevant cells in H, we randomly choose k thereof and remove them from H to create H_4.

2.3 Augmenting the Training Dataset

We propose to fine-tune models on the training set augmented with noisy data. We extend the original ToTTo training set $\mathcal{D} = \{(T, S, H)\}_{j=1}^{|\mathcal{D}|}$ with data instances with user noisy cell selections. Specifically, we replace data instances with clean cell selections H in $\mathcal{D}$ with corrupted data instances with noisy cell selections H_i. This results in a training set $\mathcal{D}_i$ consisting of noisy cell selections of noise type i. The final training set $\mathcal{D}_{final}$ contains both clean and corrupted data instances, its size is 603,805 (5 times the size of the original training set), and it is defined as $\mathcal{D}_{final} = \mathcal{D} \cup \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 \cup \mathcal{D}_4$. We set $k = 1$ for creating data instances of type Noise 1, Noise 2, and Noise 3. This is because the average number of highlight cells in ToTTo dataset is small (3.55). To create instances of type Noise 4, we remove all irrelevant cells found in H.

2.4 Robustness via Sequence Level Training

Inspired by PlanGen (Su et al., 2021), to further enhance the robustness of table-to-text models on clean and noisy cell selections, we further fine-tune model parameters with Reinforcement Learning (RL) (Williams, 1992). Formally, given an input data pair $\{T, S, H\} \in \mathcal{D}_{final}$ and a sampled output sequence $S^{'} = (S^{'}_1, ..., S^{'}_{|S^{'}|})$, the RL training objective is formulated as:

$$\mathcal{L}_{\mathcal{RL}} = -R(S, S^{'}) \sum_{i=1}^{|S^{'}|} log\, P \left(S^{'}_i \,\middle|\, S^{'}_{<i}, E(T, H) \right) \quad (1)$$

where $E(\cdot)$ denotes the encoder module of a table-to-text generator. The reward function $R(S, S^{'})$ measures the similarity between the reference text and the text generated by the model; it is formulated as $R(S, S^{'}) = B(S, S^{'})$ where $B(\cdot, \cdot)$ is the BLEU score (Papineni et al., 2002). By doing this, we make the outputs of both clean and noisy cell selections to be more similar to the reference texts. This implicitly improves the similarity between outputs of clean and noisy cell selections.

Model	Clean	Noise Avg.	Noise Var.	#Param
BART-BASE (clean)	47.8	44.0	9.09	**141M**
BART-LARGE (clean)	48.6	43.9	14.43	408M
BART-BASE ($\mathcal{D}_{final}$)	48.5	48.03	0.16	**141M**
BART-BASE + RL ($\mathcal{D}_{final}$)	49.2	**48.85**	0.14	**141M**
BART-LARGE ($\mathcal{D}_{final}$)	49.1	48.16	0.69	408M
BART-LARGE + RL ($\mathcal{D}_{final}$)	**49.6**	48.75	0.60	408M

Table 1: BLEU scores on clean and noisy development sets. Average BLEU score across the four noisy development sets (Noise Avg.). Variance of BLEU scores across the four noisy development sets (Noise Var.). Model parameters (#Param). The attribute in parenthesis indicates the dataset used for model fine-tuning.

2.5 Table-to-Text Generation Models

Our models are based on BART (Lewis et al., 2020). We fine-tune them for user controlled table-to-text generation as follows. Given a training data pair $\{T, S, H\}$, the fine-tuning process proceeds in two stages. The first stage fine-tunes the model with a conventional conditional language modelling training objective:

$$\mathcal{L}_{\mathcal{LM}} = -\sum_{i=1}^{|S|} log\, P \left(S_i \,\middle|\, S_{1:i-1}, E(T, H) \right) \quad (2)$$

where E denotes the encoder of the table-to-text generator. The second stage further adjusts model parameters by using $\mathcal{L}_{mix} = \mathcal{L}_{\mathcal{LM}} + \mathcal{L}_{\mathcal{RL}}$.

3 Experimental Results

Implementation details for our table-to-text generation models can be found in Appendix B. We use the same hyperparameters as the baseline in the ToTTo (Parikh et al., 2020).

As shown in Table 1 (detailed results per Noise type are given in Appendix A), when using the training scheme with clean cell highlights, the average BLEU score of **BART-BASE (clean)** drops from 47.8 to 44 when tested on noisy cell selections. Similar trend can be seen for **BART-LARGE (clean)** with a BLUE score drop from 48.6 to 43.9. In addition, the "Noise Variance" of **BART-BASE (clean)** and **BART-LARGE (clean)** is large, indicating that these models are not stable (or robust) to different types of noisy cell selections. All this suggests that a training scheme with carefully selected cells alone results in systems that perform poorly in practical scenarios with user interactions.

In contrast, we observe that our proposed learning scheme makes generators achieve better performance both on clean and noisy cell selections. On

Kosuke Matsuura
Section Title: IndyCar Series

Year	Team	14	16	Rank	Points
2004	Super Aguri Fernandez Racing	CHI Ret	TX2 Ret	14th	280
2005	Super Aguri Fernandez Racing	SNM 6	WGL 6	14th	320

Reference: In 2005, Kosuke Matsuura again drove for Super Aguri Fernandez Racing, and again finished 14th with a best place finish of 6th in the two races.

Ours: In 2005, Kosuke Matsuura drove for Super Aguri Fernandez Racing in the IndyCar Series and finished 14th in points.

Baseline: In 2005, Kosuke Matsuura drove for Super Aguri Fernandez Rac- -ing and finished 14th in the WGL 6 and 280 points.

Asian Beach Games
Section Title: List of Asian Beach Game

Edition	Year	City	Start Date	End Date
IV	2014	Phuket	14 November	23 November
V	2016	Da Nang	24 September	3 October
VI	2020	Sanya	24 November	5 December

Reference: The last Asian Beach Games was held in Danang, Vietnam from 24 September to 3 October 2016, while the next will be held in 2020 in Sanya, China, the first to breakaway from the 2-year cycle.

Ours: The Asian Beach Games are scheduled to be held in Da Nang, Vietnam from September 24 to October 3, 2016 and in Sanya, China in 2020.

Baseline: The Asian Beach Games were held from 2014 to 2016 in Da Nang, Vietnam and from 3 October to 3 October 2020 in Sanya, China.

List of rulers of Brittany
Section Title : House of Montfort

Name	Birth	Death
Peter II the Simple (Pêr II) 1450–1457	7 July 1418	22 September 1457 Nantes aged 41
Arthur III the Justicier (Arzhur III) 1457–1458	24 August 1393	26 December 1458 Nantes aged 65

Reference: At the very end of his life, Arthur III became duke of Brittany, succeeding Peter II.

Ours: Arthur III the Justicier was Duke of Brittany from 1457 until his death in 1458, succeeding Peter II the Simple.

Baseline: Arthur III (26 December 1458) was Duke of Brittany from 1450 to his death.

Iain Glein
Section Title: Awards and nominations

Year	Title	Award	Category	Result
1990	Silent Scream	Silver Bear	Best Actor	Won

Reference: In 1990, Glen won the Silver Bear for the Best Actor in the Silent Scream.

Ours: In 1990, Iain Glen won the Silver Bear for Best Actor for Silent Scream.

Baseline: In 1990, Iain Glen received the Silver Bear for Best Actor for Silent Scream.

Figure 2: Model outputs for synthetic noisy cell selections of type Noise 1 (left top) and Noise 2 (left bottom), and for user noisy cell selections from the human study of type Noise 3 (right top) and Noise 4 (right bottom) .

	Model	**FL**	**FA**	**CC**
clean	BART-LARGE (clean)	0.83	0.83	0.89
	BART-LARGE + RL ($\mathcal{D}_{final}$)	0.88	0.89	0.93
Noisy	BART-LARGE (clean)	0.80	0.81	0.87
	BART-LARGE + RL ($\mathcal{D}_{final}$)	0.89	0.91	0.91

Table 2: Results of Human Evaluation. Percentage of outputs perceived as Fluent (FL), Faithful (FA), and better Covering selected Cells (CC).

Method	Overall BLEU	PARENT	BLEURT
NCP	19.2	29.2	-0.576
Pointer Generator	41.6	51.6	0.076
Bert-to-Bert	44.0	52.6	0.121
LATTICE	48.4	58.1	0.222
T5-3B	**49.5**	58.4	0.230
PlanGen	49.2	58.7	**0.249**
Ours	49.3	**58.8**	0.235

Table 3: ToTTo test set results. All reported results can be found in the ToTTo leaderboard.

clean cell selections (ToTTo original development set), the model trained using the proposed learning scheme **BART-BASE** ($\mathcal{D}_{final}$) outperforms the model using the same pre-trained model but fine-tuned with the standard learning scheme **BART-BASE (clean)** by 0.7 BLEU scores. On noisy cell selections, **BART-BASE** ($\mathcal{D}_{final}$) outperforms **BART-BASE (clean)** by 4.03 BLEU points on average. In addition, **BART-BASE** ($\mathcal{D}_{final}$) has a small "Noise Variance" score across four noisy and one clean development sets, suggesting that the proposed learning scheme can make controlled table-to-text generators more robust and less sensitive to various types of noisy cell selections. Fine-tuning with RL, **BART-BASE + RL** ($\mathcal{D}_{final}$), can further boost models' performance.

In Appendix A we provide additional experiments on ablation results on the contribution of each Noise dataset, training with a subset of $\mathcal{D}_{final}$ (i.e., training with one fifth of the data also improves robustness), and evaluating on cases with different amount of noise (i.e., our approach generalises better to cases with higher values of k).

To gain insights on how the improvements are perceived in generated descriptions, we conduct a human evaluation. We follow the setup described in (Parikh et al., 2020). We sample 100 development instances and have five human judges (voluntary MSc level students fluent in English) to annotate them across three criteria. **Fluency** (users select amongst *Fluent*, *Mostly Fluent*, and *Not Fluent*; we report the percentage of outputs annotated as *Fluent*; **Faithfulness** (a candidate sentence is considered to be faithful if all the information in it is supported by the highlight cells and metadata of the table; we report the percentage of outputs that users annotate as faithful); and **Covered Cells** (the percentage of highlighted cells that the candidate sentence covers; we report average percentage of covered cells across all sampled instances). Table 2 shows that judges find outputs by the model variants fine-tuned with the proposed regime more faithful, fluent and with better cell coverage.

We choose the best performing model, **BART-LARGE + RL ($\mathcal{D}_{final}$)**, fine-tuned with the proposed approach and compare it with state-of-the-art models on the ToTTo test set. These are NCP (Puduppully et al., 2019), Pointer-Generator (See et al., 2017), Bert-to-Bert (Parikh et al., 2020), and T5-3B (Raffel et al., 2020), LATTICE (Wang et al., 2022), and PlanGen (Su et al., 2021). Table 3 shows overall results (detailed overlap/non-overlap results are provided in Appendix A). Our model performs in par with T5-3B and PlanGen despite the fact that the first one has more parameters and the second one posses a dedicated planning step.

Figure 2 shows two instances of synthetic noisy cell selections of type Noise 1 (i.e., accidentally selected random cell not related to the exploration intention) and type Noise 2 (i.e., random criteria for header selection); and two instances of user noisy cell selection from the human study of type Noise 3 (i.e., highlight *2014* semantically close to cells in the exploratory intention) and Noise 4 (i.e., *won* is not highlighted). Cells in yellow indicate original highlights from the ToTTo dataset and those in orange are noisy selections. In both cases, the outputs produced by the model fine-tuned with the proposed regime are not affected by noise and show better coverage, factual accuracy, and lexicalisation. This illustrates human evaluation preferences.

4 Conclusion

We study the performance of user controlled table-to-text generation. We show that standard training schemes with only carefully selected cells causes poor robustness of generators in practice when confronted with user noisy cell selections. To address this, we introduce a training scheme with simulated user noisy cell selections. Experimental results show that generators optimized with our proposed scheme can achieve better performance on both clean and noisy cell selections. In the future, it would be interesting to investigate how to apply our approach to other data-to-text datasets to improve model generalisation.

5 Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (award number 681760).

Limitations

We create synthetic data simulating real users interactions (i.e., user cell selections on a table). However, the automatic noise generation method does not cover all possible user interactions and may fail to exactly reproduce them in some cases. For example, our process for creating Noise 3 randomly highlights cells in the same row/column as a reference highlighted cell. However, the probability distribution of a user highlighting a cell around a reference highlighted cell is not always uniform, but in some cases based on some reasoning process about the concerned cells. In the future, it would be interesting to investigate how to simulate this reasoning process to predict where the user is likely to highlight cells. Nevertheless, the set of noise types that we propose in this work shows that models trained only on cleaned data are brittle.

References

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*, pages 97–102.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

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*, pages 7871–7880.

Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic Construction of Evaluation Suites for Natural Language Generation Datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. (NeurIPS 2021).

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*, pages 311–318.

Ankur 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)*, pages 1173–1186.

Laura Perez-Beltrachini and Mirella Lapata. 2018. Bootstrapping generators from noisy data. In *North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana. Association for Computational Linguistics. (NAACL 2018).

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6908–6915.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. *arXiv preprint arXiv:2205.03972*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

A Detailed and Ablation Results

Table 4 provides detailed results for the different model variants **(clean)** and **($\mathcal{D}_{final}$)** evaluated on different development sets with different types of noise (cf., Table 1 in Section 3). Table 5 provides detailed results comparing our model **BART-LARGE + RL** ($\mathcal{D}_{final}$) with other state-of-the-art methods in ToTTo's leaderboard (cf., Table 3 in Section 3).

We conduct an ablation study to investigate the impact of each type of noise in $\mathcal{D}_{final}$ (see Section 2.2). Specifically, we remove one of the four noise types at a time from $\mathcal{D}_{final}$, then train the **BART-BASE** model using the remaining data. This study shows that all types of user noisy cell selections help to improve performance and robustness (Table 6).

We construct corrupted ToTTo development datasets with *different amount of noise* (i.e., different number k of noisy cells) added to each original input highlighted cells. In the ToTTo dataset, there are on average 3.5 highlighted cells for each table; when $k = 3$, the injected noise has roughly the same proportion as the original highlight cells. We then examine BLEU scores for **BART-BASE** trained with our approach and the baseline on these noisy development sets. As shown in Table 7, performance drops significantly as more noise is injected, from 47.8 when $k = 0$ (clean) to 34.8 when $k = 3$, for the model trained only on clean cell selections, **BART-BASE** (clean). It also indicates that the models trained with our proposed method, **BART-BASE** **($\mathcal{D}_{final}$)** and **BART-BASE + RL** **($\mathcal{D}_{final}$)**, can reduce this performance drop.

We also combine all noise types with clean data for training in a way that the resulting dataset has the same size as the original clean dataset. Specifically, we randomly divide the original dataset into five equal parts and replace four of them each by a different type of noisy data subset; one of the parts is not replaced (i.e., one part of the original clean set is kept). We merge these five parts together and call this the mixed dataset $\mathcal{D}_{mix}$. Results in Table 8 indicate that training the model on a substantially smaller subset of clean and noisy data (i.e., a subset of $\mathcal{D}_{final}$) still yields comparable performance on clean data and significant better performance on noisy data.

Model	Clean Dev set	Noise1 Dev set	Noise2 Dev set	Noise3 Dev set	Noise4 Dev set	Noise Average	Noise Variance	#Param
BART-BASE (clean)	47.8	40.6	45.6	42.5	47.3	44	9.087	**141M**
BART-LARGE (clean)	48.6	39.8	46.1	41.7	48	43.9	14.433	408M
BART-BASE ($\mathcal{D}_{final}$)	48.5	47.7	48.6	47.9	47.9	48.025	0.156	**141M**
BART-BASE + RL ($\mathcal{D}_{final}$)	49.2	**48.6**	49.4	**48.8**	48.6	**48.850**	0.143	**141M**
BART-LARGE ($\mathcal{D}_{final}$)	49.1	46.9	48.6	47.6	48.6	48.16	0.689	408M
BART-LARGE + RL ($\mathcal{D}_{final}$)	**49.6**	47.9	**49.7**	48.4	**49.0**	48.75	0.603	408M

Table 4: BLEU scores of models on clean and noisy ToTTo development set. Average BLEU score across the four noisy development sets (Noise Avg.). Variance of BLEU scores across the four noisy development sets (Noise Var.). #Param denotes the total number of parameters in the model. The attribute in parenthesis indicates the training data we use for training the model. For (clean), models are trained on clean ToTTo training set (i.e. using $\mathcal{D}$). For ($\mathcal{D}_{final}$), the noise-augmented training set described in section 2.3 is applied. For '+RL', the Reinforcement Learning algorithm described in section 2.4 is applied.

Method	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
PlanGen	49.2	58.7	**0.249**	56.9	62.8	**0.371**	**41.5**	**54.6**	**0.126**
Ours	49.3	**58.8**	0.235	57.1	**63.4**	0.358	**41.5**	54.1	0.112

Table 5: ToTTo test set results. All reported results can be found in the ToTTo leaderboard.

Training Data	Clean Dev	Noise1 Dev	Noise2 Dev	Noise3 Dev	Noise4 Dev	Noise Avg	Noise Var
$\mathcal{D}_{final}$	48.5	**47.7**	**48.6**	**47.9**	47.9	**48.025**	0.156
$\mathcal{D}_{final} - \mathcal{D}_1$	48.5	47.3	48.4	47.6	47.8	47.775	0.216
$\mathcal{D}_{final} - \mathcal{D}_2$	**48.6**	47.6	48.5	**47.9**	**48.1**	**48.025**	**0.143**
$\mathcal{D}_{final} - \mathcal{D}_3$	48.5	47.6	**48.6**	47.8	47.9	47.975	0.189
$\mathcal{D}_{final} - \mathcal{D}_4$	48.3	**47.7**	**48.6**	**47.9**	47.4	47.900	0.260

Table 6: BLEU scores for **BART-BASE** trained on different training data and evaluated on different development sets. Noise Avg denotes the average BLEU scores on all noisy development sets. Noise Var denotes the variance of BLEU scores on noisy development sets.

Model	clean	$k=1$	$k=2$	$k=3$
BART-BASE (clean)	47.8	42.7	38.1	34.8
BART-BASE ($\mathcal{D}_{final}$)	48.5	48.1	45.9	42.3
BART-BASE + RL ($\mathcal{D}_{final}$)	**49.2**	**48.8**	**46.4**	**42.9**

Table 7: BLEU scores on input cell highlights with different amounts of noise (development set). k denotes the amount of noise added to the original data point (higher k means more noisy cell highlights are added).

Dev/Train	Clean	$\mathcal{D}_{mix}$	$\mathcal{D}_{final}$
Clean	47.8	47.3	48.50
Noise Avg.	44.0	46.8	48.03

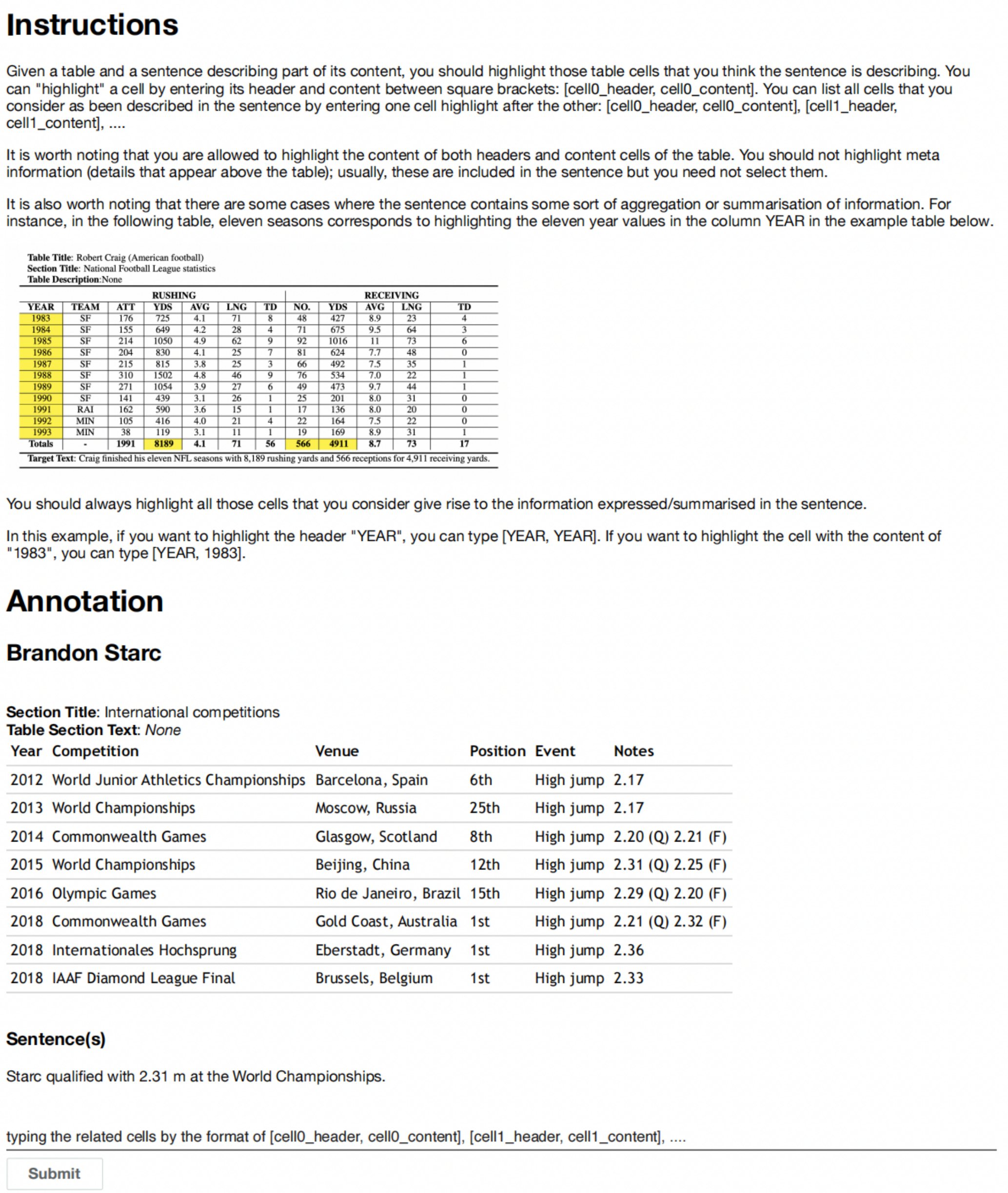
Table 8: BLEU scores of **BART-BASE** trained on the original dataset, the noise augmented dataset ($\mathcal{D}_{final}$), and a smaller dataset ($\mathcal{D}_{mix}$). Evaluation is on clean and Noise development sets.

B Implementation Details

The examined models are based on the Huggingface Library (Wolf et al., 2020) with default model hyperparameters provided by the Library. We fine-tune BART (Lewis et al., 2020) using the proposed learning scheme. We use the Adam (Kingma and Ba, 2014) optimizer, with a learning rate of $2e^{-5}$ and a batch of size 32. We fine-tune with the $\mathcal{L}_{\mathcal{LM}}$ objective for 100k steps and $\mathcal{L}_{mix}$ for 50k steps.

C Human Study Interface

Figure 3 shows the Amazon Mechanical Turk interface, instructions and annotation form, we use for the human study described in Section 2.1.



Instructions

Given a table and a sentence describing part of its content, you should highlight those table cells that you think the sentence is describing. You can "highlight" a cell by entering its header and content between square brackets: [cell0_header, cell0_content]. You can list all cells that you consider as been described in the sentence by entering one cell highlight after the other: [cell0_header, cell0_content], [cell1_header, cell1_content], ...

It is worth noting that you are allowed to highlight the content of both headers and content cells of the table. You should not highlight meta information (details that appear above the table); usually, these are included in the sentence but you need not select them.

It is also worth noting that there are some cases where the sentence contains some sort of aggregation or summarisation of information. For instance, in the following table, eleven seasons corresponds to highlighting the eleven year values in the column YEAR in the example table below.

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

		RUSHING					RECEIVING				
YEAR	**TEAM**	**ATT**	**YDS**	**AVG**	**LNG**	**TD**	**NO.**	**YDS**	**AVG**	**LNG**	**TD**
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Target Text: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

You should always highlight all those cells that you consider give rise to the information expressed/summarised in the sentence.

In this example, if you want to highlight the header "YEAR", you can type [YEAR, YEAR]. If you want to highlight the cell with the content of "1983", you can type [YEAR, 1983].

Annotation

Brandon Starc

Section Title: International competitions
Table Section Text: *None*

Year	Competition	Venue	Position	Event	Notes
2012	World Junior Athletics Championships	Barcelona, Spain	6th	High jump	2.17
2013	World Championships	Moscow, Russia	25th	High jump	2.17
2014	Commonwealth Games	Glasgow, Scotland	8th	High jump	2.20 (Q) 2.21 (F)
2015	World Championships	Beijing, China	12th	High jump	2.31 (Q) 2.25 (F)
2016	Olympic Games	Rio de Janeiro, Brazil	15th	High jump	2.29 (Q) 2.20 (F)
2018	Commonwealth Games	Gold Coast, Australia	1st	High jump	2.21 (Q) 2.32 (F)
2018	Internationales Hochsprung	Eberstadt, Germany	1st	High jump	2.36
2018	IAAF Diamond League Final	Brussels, Belgium	1st	High jump	2.33

Sentence(s)

Starc qualified with 2.31 m at the World Championships.

typing the related cells by the format of [cell0_header, cell0_content], [cell1_header, cell1_content], ...

Submit

Figure 3: The Amazon Mechanical Turk interface, instructions and annotation form, we use for the human study described in Section 2.1.