A Tree-to-Sequence Model for Neural NLG in Task-Oriented Dialog

Jinfeng Rao, Kartikeya Upasani, Anusha Balakrishnan,
Michael White, Anuj Kumar, Rajen Subba
Facebook Assistant
{raojinfeng,kart,anushabala,,mwhite14850,anujk,rasubba}@fb.com

Abstract

Generating fluent natural language responses from structured semantic representations is a critical step in task-oriented conversational systems. Sequence-to-sequence models on flat meaning representations (MR) have been dominant in this task, for example in the E2E NLG Challenge. Previous work has shown that a tree-structured MR can improve the model for better discourse-level structuring and sentence-level planning. In this work, we propose a tree-to-sequence model that uses a tree-LSTM encoder to leverage the tree structures in the input MR, and further enhance the decoding by a structure-enhanced attention mechanism. In addition, we explore combining these enhancements with constrained decoding to improve semantic correctness. Our method not only shows significant improvements over standard seq2seq baselines, but also is more data-efficient and generalizes better to hard scenarios.

1 Introduction

Generating fluent natural language responses from structured semantic representations is crucial to building engaging and effective task-oriented dialog systems. Neural approaches for natural language generation (NNLG), particularly sequence-to-sequence approaches, have achieved promising results and were dominant in the recent E2E Challenge. Most of these approaches are built on flat meaning representations (MR) that use key-value pairs to capture attributes to be conveyed in responses. However, coupled with such flat MRs, current NNLG methods still struggle with 1) reliably performing sentence-level planning and discourse-level structuring (Reed et al., 2018); 2) avoiding generating semantic errors like hallucinated content (Dušek et al., 2018, 2019); and 3) generalizing to hard inputs (Wiseman et al., 2017).

To help overcome these drawbacks, Balakrishnan et al. (2019) propose a novel tree-structured meaning representation to gain better control of the discourse structure and content in generated utterances. Their proposed tree-structured MRs consist of three sets of non-terminal tokens: argument, dialog act and discourse act. A dialog act is a minimum atomic unit that contains a few arguments to be expressed in an utterance, while discourse acts define the relationship between dialog acts.

An example of their tree-structured MR for the weather domain is provided in Table 1, along with a flat MR and human reference. We also add a reference annotated with the tree-structured MR in the last row. The tree-structured MR provides much better controllability to a live task-oriented dialog system, where developers can easily inject external knowledge into a rule-based response planner to specify the relationship between multiple dialog acts (e.g., rainy is the opposite to sunny), and the grouping of arguments in a dialog act is possible. These consideration have been shown to be critical to user perceptions of quality and naturalness (Lemon et al., 2004; Carenini and Moore, 2006; Walker et al., 2007; White et al., 2010; Demberg et al., 2011).

In their Seq2Seq model, Balakrishnan et al. (2019) treat the tree-structured MR as just a sequence of tokens, ignoring the inherent tree structure (though this structure is taken into account in constrained decoding). We aim to examine the hypothesis that a better representation of the input tree structures could lead to better generalizability of the model and enhance semantic correctness. Therefore, we propose a tree-to-sequence model that uses a tree-based encoder to better represent the tree-structured MRs, and a structure-enhanced decoder to further incorporate contextual information in decoding.

Our contributions are summarized as follows:

Proceedings of The 12th International Conference on Natural Language Generation, pages 95–100,
Tokyo, Japan, 28 Oct - 1 Nov, 2019. ©2019 Association for Computational Linguistics

Reference	It'll be sunny throughout this weekend. The high will be in the 60s, but expect temperatures to drop as low as 43 degrees by Sunday evening. There's also a chance of strong winds on Saturday morning.
Flat MR	`condition1[sunny] date_time1[this weekend] avg_high1[60s] low2[43] date_time2[Sunday evening] chance3[likely] wind_summary3[strong] date_time3[Saturday morning]`

Our MR	`INFORM [ condition[sunny], date_time_range[ colloquial[this weekend ] ] ]`
	`CONTRAST [`
	`   INFORM [ avg_high[60s] date_time[ [colloquial this weekend ] ] ]`
	`   INFORM [ low[43] date_time[ week_day[Sunday] colloquial[evening] ] ]`
	`]`
	`INFORM [ chance[likely], wind_summary[heavy], date_time[ week_day[Saturday] colloquial[morning] ] ]`

Annotated Reference	[INFORM It'll be [condition sunny] throughout [date_time_range colloquial[this weekend]]. [CONTRAST [INFORM The high will be in the [avg_high 60s]]] , [INFORM but expect temperatures to drop as low as [avg_low 43 degrees] by [date_time [week_day Sunday] [colloquial evening]]]. [INFORM There's also [chance a chance of] [wind_summary strong winds] on [date_time [week_day Saturday] [colloquial morning]] .]

Table 1: Sample flat MR with reference compared against tree-structured MR. The last row shows an annotated reference with the tree-structured MR. Nodes in blue are all children of the root node of the tree.

- We propose a tree-to-sequence (tree2seq) model to better leverage the inherent structures in the tree-based MRs. Coupled with the constrained decoding technique from (Balakrishnan et al., 2019), we explore whether combining better learning and decoding methods yields the best performance.

- Extensive evaluations on conversational weather and E2E datasets (Dušek et al., 2019) show that the tree2seq model can significantly improve semantic correctness. Analysis further shows that tree2seq is more data-efficient and generalizes better to hard scenarios.

2 Related Work

Several previous works have focused on adding planning steps to neural NLG architectures or employed non-sequential encoders. Puduppully et al. (2019) add a content planning step where a set of input database records are mapped to an ordered list of selected records; however, their approach does not employ hierarchical content plans as in our approach. Moryossef et al. (2019) add a symbolic text planning step where facts are grouped and ordered in the input; in contrast to our work though, their approach uses standard Seq2Seq models for realization and leaves no ordering choices to the model. Previous work on AMR and WebNLG (Beck et al., 2018; Song et al., 2018; Marcheggiani and Perez-Beltrachini, 2018) has demonstrated improvements over Seq2Seq models by using graph-to-sequence models; while similar in principle, these works do not explore the use of hierarchical content plans as intermediate structures and do not experiment with constrained decoding.

Elder et al. (2019) propose using an intermediate representation motivated by a universal dependency tree, and find that this greatly improves performance. However, their approach is still Seq2Seq-based and can't explicitly model the tree structures. Similar to our approach, Eriguchi et al. (2016) use a tree-to-sequence model for machine translation, but here we focus on NLG and use different tree encoder and constrained decoding techniques.

3 Tree-to-Sequence Model

3.1 Tree-Based Encoder

The input to our model is a tree-structured MR, and the output is an annotated reference, e.g., the last row in Table 1. Having annotated non-terminal tokens in the output allows us to check whether all arguments are expressed in output following the input tree structures.

We represent each token in the input MR as a tree node, using the tree structure to compute the hidden state of the k-th parent node $\mathbf{h}_k^p$ as a function of its child states $\{\mathbf{h}_k^{c_1}, ..., \mathbf{h}_k^{c_N}\}$:

$$\mathbf{h}_k^p = f_{\text{tree}}(\{\mathbf{h}_k^{c_1}, ..., \mathbf{h}_k^{c_N}\})$$

where N is the number of children for k-th node and f_{tree} is a non-linear function. We implemented a variant of the N-ary TreeLSTM by (Tai et al., 2015) as our tree encoder.

Since trees can have completely different layouts, it's hard to train and do inference with tree inputs in parallel. We propose an iterative bottom-up traversal algorithm to support batch forward and backward with tree inputs. Given a batch of trees, we first extract all the leaf nodes and update their states in a batch manner. Then we iteratively update the states of non-leaf nodes if all of their children nodes have been processed. As nodes can have different number of children nodes, we padded non-leaf nodes to have the same number of children nodes (i.e., N) for batch processing. Overall, the batch calculation ends up with 5-10X speedup compared to single-tree forward, allowing us to train on large datasets.[1]

3.2 Structure-Enhanced Decoder

The tree-based encoder returns a list of hidden states $\{\mathbf{h}_1, ..., \mathbf{h}_K\}$, where K is the length of source sequence. We first initialize the initial decoder state $\mathbf{s}_1$ as its root hidden state:

$$\mathbf{s}_1 = \mathbf{h}_{\text{root}}$$

In a standard attentional seq2seq (Bahdanau et al., 2014), $\alpha_j(k)$ denotes the attention score between j-th target state $\mathbf{s}_j$ and k-th source state $\mathbf{h}_k$. Then a weighted sum over source hidden states are calculated as $\mathbf{d}_j = \sum_k \alpha_j(k)\mathbf{h}_k$, and is used for updating the context state as follows:

$$\hat{\mathbf{s}}_j = \tanh(\mathbf{W}_d \cdot [\mathbf{s}_j; \mathbf{d}_j] + \mathbf{b}_d) \tag{1}$$

where $[\mathbf{s}_j; \mathbf{d}_j]$ is a concatenation of hidden state $\mathbf{s}_j$ and $\mathbf{d}_j$. Next $\widehat{\mathbf{s}_j}$ is used for predicting the j-th target token:

$$P(y_j|\mathbf{y}_{<j}, x) = \text{softmax}(\mathbf{W}_s \cdot \widehat{\mathbf{s}_j} + \mathbf{b}_s)$$

However, the above decoding procedure doesn't take the tree structures into account. We adopted the input feeding approach (Eriguchi et al., 2016) by modifying equation (1) to feed the previous unit $\mathbf{s}_{j-1}$ to update the j-th context state:

$$\widehat{\mathbf{s}_j} = \tanh(\mathbf{W}_d \cdot [\mathbf{s}_j; \mathbf{d}_j; \mathbf{s}_{\hat{j-1}}] + \mathbf{b}_d)$$

The input feeding approach allows us to enrich the contextual information when predicting the current token, in particular because $\mathbf{s}_{\hat{j-1}}$ is often the parent state of j-th node (given that the output tree structures are linearized to a sequence of words).

3.3 Constrained Decoding

Balakrishnan et al. (2019) propose a constrained decoding approach that derives constraints from the input tree structure to be enforced during decoding. In the beam search process, if a predicted non-terminal token violates the input MR structure, then the token is rejected. This allows beam search to explore more valid hypotheses with the same beam size. Their experiments show that constrained decoding can significantly improve the semantic correctness of generated responses by avoiding missing/repeating arguments and reducing hallucinated content while also enforcing desired groupings. (See their paper for further details on how the constraints are enforced.)

Though constrained decoding yields promising results in Balakrishnan et al.'s experiments, it's worth observing that constrained decoding does not affect the training process, which means that it doesn't help with generalization and relies on a strong base model. Therefore, we experiment with combining our tree-to-sequence model with constrained decoding, in order to determine whether the two methods work better in combination.

4 Experiments

4.1 Setup

Datasets: We conducted experiments on both the enriched E2E dataset and the weather dataset from (Balakrishnan et al., 2019).

Models We consider both Seq2Seq-based models and our proposed Tree2Seq models in our experiments. All Seq2Seq models use an LSTM-based encoder and decoder, with attention, while the Tree2Seq models have the architecture described in Section 3.

- **S2S**: Standard S2S-TREE model proposed in (Balakrishnan et al., 2019). This is a Seq2Seq model in which the input is a linearized text representation of the MR, while the output is an annotated response (example in Table 1).
- **S2S-CONSTR**: This is the S2S-CONSTR model proposed in (Balakrishnan et al., 2019). This is identical in architecture to the S2S model, and differs only in the decoding step, where constrained decoding is applied to ensure semantic correctness.
- **T2S**: Our proposed model, with tree-based encoding and structure-enhanced decoding.
- **T2S-CONSTR**: Has the same architecture as T2S, but with constrained decoding applied to the decoder to ensure semantic correctness.

Metrics We consider both automatic metrics and human evaluation results. For automatic metrics, we evaluate on following automatic metrics:

[1] On the E2E dataset, the batchized tree2seq model takes 20 and 2 minutes every epoch in training and testing.

Model	E2E					Weather				
Metric	BLEU	TreeAcc		Gram.	Corr.	BLEU	TreeAcc		Gram.	Corr.
	-	NoDisc	Disc	-	-	-	NoDisc	Disc	-	-
S2S	74.58	99.68	95.28	93.59	83.85	76.75	96.62	83.30	94.17	87.40
S2S-Constr	74.69	**99.89**	97.78	94.33	**85.89**	77.45	98.52	91.61	94.20	90.40
Our Approaches										
T2S	**74.75**	**99.89**	96.96	94.83	84.66	**77.86**	97.1	88.80	**94.55**	89.75
T2S-Constr	74.63	99.84	**98.60**	**94.68**	85.68	77.82	**99.11**	**94.13**	94.14	**91.84**

Table 2: Results on E2E and Weather datasets. All metrics are percentages.

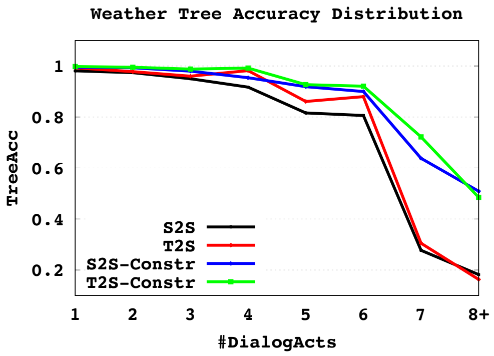

Figure 1: Tree Accuracy Distribution on Weather

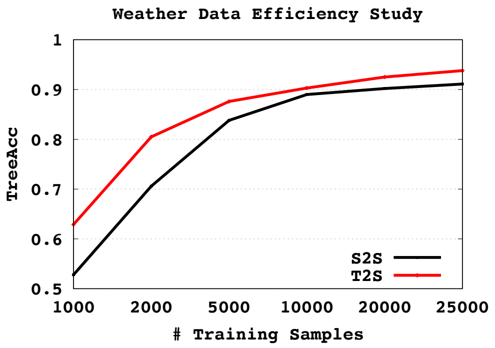

Figure 2: Data Efficiency Study on Weather

1) **BLEU-4** (Papineni et al., 2002); 2) **Tree accuracy** (Balakrishnan et al., 2019), which is a binary metric to indicate whether the tree structure in the prediction matches that of the input MR exactly.

For human evaluation, annotators rate model responses in a **binary** scale on two dimensions:

- **Grammaticality** (`Gram`): Our evaluation guidelines included considerations for proper subject-verb agreement, word order, grammatical completeness, etc..
- **Correctness** (`Corr`): Measures *semantic correctness* of the responses. Our guidelines considered sentence structure, contrast, hallucinations (incorrectly included attributes), and missing attributes. We asked annotators to evaluate model predictions against the reference rather than the MR.

Our human evaluation was conducted in a double-blind setting, in which two annotators independently provide ratings for each response, and a third annotator resolves any disagreements between the two. The disagreement rate is 20.7% for the weather dataset and 23.2% for the E2E dataset.

4.2 Main Results

Table 2 shows the main results. For the tree accuracy metric, we report the numbers on two disjoint subsets: discourse subset (column Disc), which contains inputs with 1+ discourse acts, and no-discourse subset (column NoDisc), which includes inputs without any discourse acts. The discourse subset is expected to be more challenging as it contains longer and more complex inputs.

From the table, we can see that all approaches are roughly comparable on BLEU scores. With tree accuracy, T2S consistently outperforms S2S in on both the discourse and no-discourse subsets, with the exception of the NoDisc subset of the E2E data, where all models are close to 100% accuracy. The margins of improvement from T2S are higher on the discourse subset, suggesting T2S is more effective on hard inputs. S2S-Constr consistently outperforms S2S and T2S, affirming the effectiveness of constrained decoding. Overall, combining the enhanced encoding and decoding methods, T2S-Constr achieves the best performance on all subsets (again, except with NoDisc tree accuracy for E2E, where ceiling performance is effectively reached).

For grammaticality, we see all approaches are comparable in E2E and Weather. Analysis show above 90% grammatical errors are because models tend to generate run-on responses that group many arguments in one sentence, without appropriate punctuation (e.g., commas). Consistent with Balakrishnan et al. (2019), we found higher tree accuracy usually corresponds to higher human judgements of semantic correctness (except with S2S-Constr and T2S-Constr for E2E). We also note that there's a noticeable gap in the E2E dataset

Column	G/T/C	Text
MR Input	-	Join [INFORM [area[riverside] name[_name_] price_range[high] eat_type[restaurant] [near _near_]] CONTRAST [INFORM [rating[1 out of 5] family_friendly_no[no]] INFORM [food_italian[_food_italian_]]]]
Annotated Reference	-	[Join [INFORM There is a [price_range high] price range [eat_type restaurant] [name name] in [area riverside] near [_near_]] [CONTRAST [INFORM It is [family_friendly not kids friendly] and [rating has low] customer ratings] but [INFORM they serve [food_italian Italian food] .]]]
S2S	1/0/0	[Join [INFORM [name _name_] is a [price_range high] priced [eat_type restaurant] in the [area riverside] area near [near _near_] .] [CONTRAST [INFORM is [family_friendly_no not children friendly] ,] but [INFORM it has a customer rating of [rating 1 out of 5] .]]]
T2S	1/0/0	[Join [INFORM [name _name_] is a [price_range high] priced [eat_type restaurant] in the [area riverside] area near [near _near_] .] [CONTRAST [INFORM it has a customer rating of [rating 1 out of 5] ,] but [INFORM is [family_friendly_no not children friendly] and serves [food_italian Italian] food.]]]
T2S-Constr	1/1/1	[Join [INFORM [name _name_] is a [price_range high] priced [eat_type restaurant] in the [area riverside] area near [near _near_] .] [CONTRAST [INFORM it is [family_friendly not children friendly] and has a [rating 1 out of 5] rating] but [INFORM it serves [food_italian Italian] food.]]]

Table 3: Sample model responses. Row 'Annotated Reference' is human-annotated reference. Column 'G/T/C' represents the value of grammaticality/tree accuracy/correctness of model predicted response.

where the tree accuracy doesn't align with the correctness numbers from human evaluation (the gap on weather is smaller). Our analysis show most correctness errors are mainly due to: 1) the compositional MR inputs were missing information in the reference which was caused by the noisy dataset creation by Balakrishnan et al. (2019); 2) some attributes caused confusions to human annotators, e.g., "20-30 pounds" can imply "a mid-priced restaurant"; 3) a legitimate amount of content hallucinations, especially in hard inputs and unseen attributes.

We also plot the tree accuracy distribution against the number of dialog acts in Figure 1. (We skip this figure for E2E dataset for space reasons, as it shows similar pattern to weather dataset.) Clearly, for smaller numbers of dialog acts (#DialogActs $<= 3$), all models perform roughly the same and almost hit 100% accuracy. But the gains of T2S is much more clear when the number of dialog acts is larger than 3. T2S-Constr is also generally better than S2S-Constr in most cases, and both are more effective for complex MRs (7 or 8+ dialog acts), where there are very few MRs (less than 0.5%) in the training set.

Data Efficiency. We set up a data efficiency experiment, in which we trained each model on increasingly larger subsets of our training set (while keeping the test set constant). Figure 2 shows the results of this experiment. Overall T2S consistently outperforms S2S, and the difference is larger with fewer training samples. This suggests that structure awareness leads to better representations and improves data efficiency.

4.3 Sample Analysis

We also provide some sample responses on E2E in Table 3. Column 'G/T/C' stands for the value of grammaticality, tree accuracy and correctness of model prediction. We obviate S2S-Constr response here as it is similar to T2S-Constr.

From the example, we can see that T2S mistakenly contrasted the family friendly and customer rating attributes, largely due to the overwhelming contrast patterns between family friendly and customer rating in training data. In addition to the contrast mistake, S2S completely ignores the attribute of serving Italian food, suggesting its poor generalization ability to rare argument (i.e., food_italian). T2S-Constr shared the first sentence with the T2S approach, but was able to correct the constrast mistake by adding structure constraints during beam search.

5 Conclusion

In this paper, we have demonstrated via experiments on two datasets that a tree-to-sequence model that leverages the inherent tree structures in input MRs can improve semantic correctness over a sequence-to-sequence model and is more data-efficient. Moreover, we have shown that the tree-to-sequence model can be coupled with a better constrained decoding method to achieve better semantic correctness than either method alone.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. To appear.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Giuseppe Carenini and Johanna D. Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170:925–952.

Vera Demberg, Andi Winterboer, and Johanna D Moore. 2011. A strategy for information presentation in spoken dialog systems. *Computational Linguistics*, 37(3):489–539.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG Challenge. *arXiv preprint arXiv:1901.11528*.

Henry Elder, Jennifer Foster, James Barry, and Alexander OConnor. 2019. Designing a symbolic intermediate representation for neural surface realization. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 65–73.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. *the 54th Annual Meeting of the Association for Computational Linguistics*.

Oliver Lemon, Johanna Moore, Mary Ellen Foster, and Michael White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of FLAIRS*. AAAI.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9, Tilburg University, The Netherlands. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics. ArXiv preprint arXiv:1904.03396.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL-02*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of AAAI*. ArXiv preprint arXiv:1809.00582.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 284–295. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Marilyn Walker, Amanda Stent, Francois Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research (JAIR)*, 30:413–456.

Michael White, Robert A. J. Clark, and Johanna D. Moore. 2010. Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, 36(2):159–201.

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