Have Your Text and Use It Too!
End-to-End Neural Data-to-Text Generation with Semantic Fidelity

Hamza Harkous
Amazon Alexa
hamza.harkous@gmail.com

Isabel Groves
Amazon Alexa
isabeg@amazon.com

Amir Saffari
Amazon Alexa
amsafari@amazon.com

Abstract

End-to-end neural data-to-text (D2T) generation has recently emerged as an alternative to pipeline-based architectures. However, it has faced challenges generalizing to new domains and generating semantically consistent text. In this work, we present DATATUNER, a neural, end-to-end data-to-text generation system that makes minimal assumptions about the data representation and target domain. We take a two-stage generation-reranking approach, combining a fine-tuned language model with a semantic fidelity classifier. Each component is learnt end-to-end without needing dataset-specific heuristics, entity delexicalization, or post-processing. We show that DATATUNER achieves state of the art results on automated metrics across four major D2T datasets (LDC2017T10, WebNLG, ViGGO, and Cleaned E2E), with fluency assessed by human annotators as nearing or exceeding the human-written reference texts. Our generated text has better semantic fidelity than the state of the art on these datasets. We further demonstrate that our model-based semantic fidelity scorer is a better assessment tool compared to traditional heuristic-based measures of semantic accuracy.

1 Introduction

Data-to-Text generation (D2T) is defined as automatically generating natural language texts from non-linguistic inputs (Reiter and Dale, 2000). Interest in this task has been driven by its applicability to specialized domains. For instance, D2T has been applied to generating weather reports (Liang et al., 2009), restaurant descriptions (Novikova et al., 2017b), and video game dialogues (Juraska et al., 2019). Recently, researchers have investigated D2T with more diverse domains to arrive at more generalizable text generation (such as works on LDC2017T10 (Knight et al., 2017) and WebNLG (Gardent et al., 2017) datasets).

Traditional approaches to D2T follow a pipeline-based methodology, dividing the problem into several sub-problems (Reiter and Dale, 2000; Gatt and Krahmer, 2018). These include content selection (which information to include in the text), text structuring (the order in which to present the data), sentence aggregation (which information goes in individual sentences), lexicalization (finding the right words and phrases to express the data), referring expression generation (selecting the words and phrases to identify domain objects), and linguistic realization (combining all the generated words and phrases into well-formed sentences).

In recent years, there has been a growing interest in going beyond pipeline-based approaches towards end-to-end (E-to-E) methods driven by recent advancements in deep learning (Lebret et al., 2016; Novikova et al., 2017b; Castro Ferreira et al., 2019; Dušek et al., 2020). Such methods can be trained with (*data,text*) tuples that can be efficiently collected at scale. In contrast, each step in pipeline-based approaches requires its own setup and training data, such as semantic alignments between sections of text and components of the meaning representation (*MR*). This makes them more costly and complex to develop and more prone to error propagation.

To date, end-to-end D2T has faced two main challenges: (1) **generalization** to unseen domains and (2) maintaining **semantic fidelity** to accurately convey the source data. In a recent comparative study, Castro Ferreira et al. (2019) found that, compared to the best pipeline-based system, E-to-E approaches based on GRU and Transformer architectures scored more than 35 BLEU points lower on unseen domains from the WebNLG dataset, and scored worst for semantic accuracy.

To address these challenges, we introduce DATATUNER, an E-to-E, domain-independent D2T system that makes no assumptions about the generated text's domain or the *MR*'s structure. DATATUNER leverages a pretrained language model and fine-grained state embeddings to achieve strong generalization. It also employs a weakly-supervised Semantic Fidelity Classifier (SFC) to detect and avoid generation errors (such as hallucination and omission). We also leverage this classifier to assess outputs from any D2T system, overcoming the limitations of existing heuristic methods for detecting semantic errors.

In this work, we deliver four main contributions across four major D2T datasets from various domains and MRs.

- We show that DATATUNER pushes the state of the art on automated metrics by significant margins, ranging from 1.2 to 5.9 BLEU points, compared to the best existing pipeline and E-to-E techniques.

- With a crowdsourcing experiment, we demonstrate that DATATUNER generates text with significantly better fluency than existing works. On two datasets, our texts are even judged to be better, on average, than human-written references.

- We show that DATATUNER improves the semantic accuracy of generated texts, with margins ranging from 5.3% to 40% as assessed by crowdsourcing workers.

- With expert annotations, we further show that our model-based semantic accuracy metric is 4.2% to 14.2% more accurate in detecting semantic errors than existing heuristic-based approaches.

We open-source the DATATUNER code at `https://github.com/amazon-research`.

2 Related Work

Pipeline vs. End-to-End Approaches: Within the pipeline-based paradigm, several studies have illustrated that breaking the D2T problem into sub-problems improves overall performance. Moryossef et al. (2019b) showed that separating planning from realization helps achieve better semantic faithfulness compared to an E-to-E neural approach on the WebNLG dataset. Castro Ferreira et al. (2019) conducted a comparative study across a variety of E-to-E and pipeline approaches with WebNLG, concluding that the latter are significantly better at generalizing to unseen domains. However, so far the E-to-E approaches in these studies have been trained from scratch on the task dataset. Our work investigates whether using a pretrained model with strong language generation capabilities raises the performance of E-to-E models.

Structured Representations of the Data: Another thread of research focuses on better encoders for meaning representation languages, exploiting their structural properties. This is particularly relevant to AMR (Damonte and Cohen, 2019; Ribeiro et al., 2019; Zhu et al., 2019; Guo et al., 2019). Damonte and Cohen (2019) showed that replacing sequential encoders with a graph encoder improves text quality as measured by BLEU and METEOR scores. Zhu et al. (2019) proposed using self-attention to better model indirectly connected AMR components. In this work, we are the first to design a system that achieves a strong performance across different data structures, ranging from slot-value pairs to graph-based *MR*. We also show that such a system can deliver significant gains compared to existing specialized systems.

Semantic Fidelity Guarantees: To improve semantic fidelity (how accurately the generated text conveys the meaning) in E-to-E architectures, one approach has been to train reverse "Text-to-Data" models (Chisholm et al., 2017; Agarwal et al., 2018). Another approach by Kedzie and McKeown (2019) used data augmentation and a reliable *MR* parser to reduce semantic errors in the generated text. Nie et al. (2019) focused on fixing training data errors via an iterative data refinement technique using a language understanding module. Nie et al. (2018) tackled the specific case where symbolic operations (e.g. numerical comparisons) are needed, augmenting the encoded input by pre-calculating these inferrable facts.

Shen et al. (2019) used techniques from computational pragmatics and modeled the generation task as a game between speakers and listeners. Despite following the generation-reranking paradigm explored previously in the data-to-text domain (Agarwal et al., 2018; Moryossef et al., 2019a; Dušek et al., 2019), and in other domains including machine translation (Shen et al., 2004), dialogue generation (Wen et al., 2015), and ASR (Morbini et al., 2012), our work has several distinctive aspects compared to previous works. First, we do not make extra assumptions, such as availability of precise *MR* parsers. Second, our system provides improvements even when the data is not the root cause of semantic errors. Third, we go beyond *encouraging* the model to avoid semantically inconsistent outputs: we aim to also *detect* with high probability when the generated text still contains such errors.

For industrial NLG applications, including in healthcare (Pauws et al., 2019) or news (Leppänen et al., 2017), identifying individual generations that are inaccurate is vital for the system to be useful in practice (Smiley et al., 2017). This error detection task has commonly relied on handwritten mappings from data values to potential realizations. Such rules were used to compute a Slot Error Rate (SER) metric (Dušek et al., 2019; Juraska et al., 2019; Moryossef et al., 2019a). For instance, Dušek et al. (2019) use SER for reranking beam elements during decoding in an attention-based sequence-to-sequence model on the Cleaned E2E dataset. Juraska et al. (2019) used the approach similarly with a transformer model on the ViGGO dataset. This technique is difficult to scale to new domains or languages, and struggles when the *MR* is not dominated by values that occur verbatim in the text (e.g. named entities). We aim to tackle that with our model-based semantic fidelity classifier.

3 Problem Description

The D2T task is formally defined as generating text T from data D that is encoded via a meaning representation *MR*. We assume that content selection is done prior to the D2T task, an assumption also made in the datasets we use. Therefore, the text T should have *semantic fidelity* by conveying all the input data, and only the input data.

3.1 Datasets

We selected the major datasets that satisfy the task definition above. Each dataset consists of (D,T) pairs with texts in English. The following describes each dataset and our preprocessing/linearization, including special tokens added (highlighted in <**bold**> below) to guide our models. We provide the datasets' statistics in Table 1 of the appendix, and we show examples from them in Figure 1.

WebNLG: In WebNLG, D is a set of 1-7 DBpedia triples which T verbalizes (Gardent et al., 2017). The test data spans 15 domains, 10 of which are seen in training. We linearize by concatenating triples, adding special tokens for 'subject', 'predicate', and 'object', and converting strings to sentence-case. For fair comparison with the state of the art, we use v1.4 from Castro Ferreira et al. (2018).

LDC2017T10: In the LDC2017T10 dataset (Knight et al., 2017), D is an Abstract Meaning Representation (AMR) graph representing "who is doing what to whom" for each sentence in T. The texts include broadcast news and weblogs. We use the preprocessing script from Ribeiro et al. (2019), without lowercasing. We merge leaves that correspond to one entity (e.g. "*United States*" below). Each role specifier is replaced with a special token.

Cleaned E2E: The Cleaned E2E dataset introduced in (Dušek et al., 2019) is an automatically cleaned version of the original E2E dataset (Novikova et al., 2017b), aiming to eliminate omissions and hallucinations in the human text by fixing the corresponding *MR*. Each *MR* consists of 3-8 slot-value pairs in the restaurant domain. We preprocess D by adding special tokens before each slot type.

ViGGO: In ViGGO (Juraska et al., 2019), D is a meaning representation with one of 9 dialogue acts (e.g. *give_opinion, suggest*) and 1-8 slot-value pairs from 14 video game attributes (e.g. NAME, GENRES). Each T is an utterance representing a dialogue turn. We add special tokens at the start and end, representing the dialog act, and before each slot type.
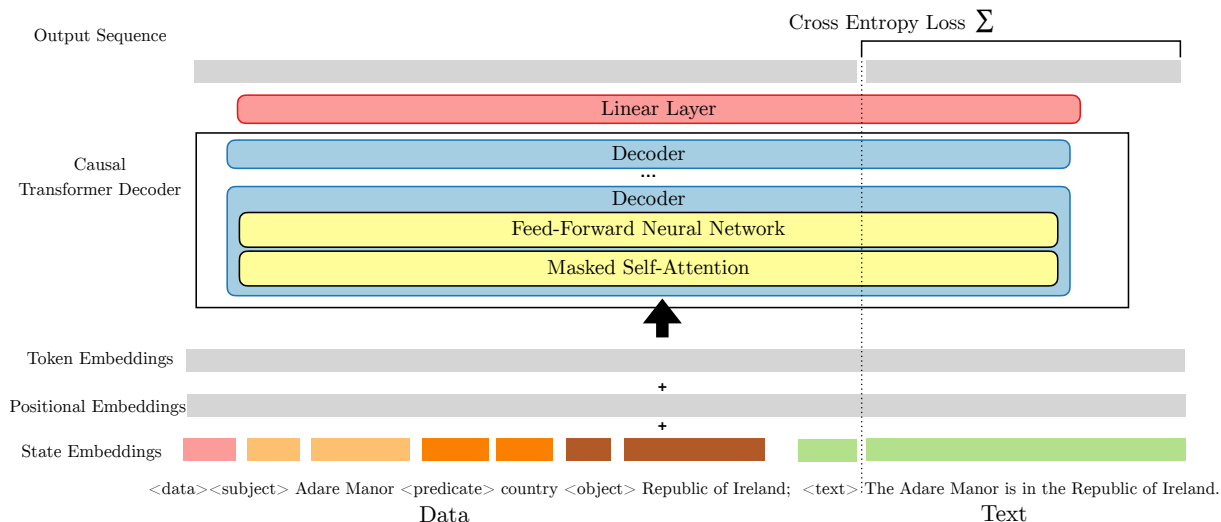
Dataset	Train Size	Validation Size	Test Size	Unique Words	% Capitalized	Dale-Chall
Cleaned E2E	33525	4299	4693	1966	29%	0.85
LDC2017T10	36521	1368	1371	5533	22%	6.49
ViGGO	5103	714	1083	2014	33%	1.02
WebNLG	18081	2260	4928	7253	63%	1.03

Table 1: Dataset text statistics

As we illustrate in Table 1, the datasets vary widely. LDC2017T10 dataset is not bounded to specific domains. Hence, although the AMR format closely describes the text, it is non-trivial to generalize from the training to test data. WebNLG covers a wide, but restricted set of domains, only a subset of which are present in the training data. However it has high lexical diversity. The number of unique words in the test set of WebNLG is 7253 (63% of them capitalized), compared to 5533 (22% capitalized) for LDC2017T10, 2014 (33% capitalized) for ViGGO, and 1966 (29% capitalized) for Cleaned E2E. Measured with the New Dale–Chall readability score (Dale and Chall, 1948), LDC2017T10 has the highest difficulty score (6.49) compared to 1.03, 0.85, and 1.02 for the WebNLG, Cleaned E2E, and ViGGO datasets respectively. In terms of quality, ViGGO was designed with the goal of perfect semantic fidelity, and Cleaned E2E was heavily filtered from the original dataset to achieve that. On the other hand, the versions we use of the other datasets have not undergone such filtering.

4 DATATUNER Architecture

We designed DATATUNER to be highly generic in order to tackle diverse meaning representations and allow D2T generators to be built for new datasets with minimal work beyond data preprocessing. At a high-level, our text generation system takes a 2-stage approach: *generation* and *reranking*. First, we fine-tune a pretrained language model on the D2T task using the task's training data. Next, we build a specialized semantic fidelity classifier trained on an automatically-generated task-specific corpus. Using these models, we construct a customized beam-search decoder that ranks candidates based on the probabilities from the language model, and, at its final stage, reranks them based on the classifier's labels.

4.1 Data-to-Text Model Fine-tuning

The fine-tuned Data-to-Text Language Model (D2T-LM) builds on the pretrained OpenAI GPT-2 model (Radford et al., 2019), a multi-layer, autoregressive language model. Each layer is a transformer decoder block (Vaswani et al., 2017) of masked multi-headed attention and a fully connected layer. We provide a full model diagram in Figure 2.

WebNLG
D= Aarhus | leaderName | Jacob_Bundsgaard
Linearized D= <**subject**> Aarhus <**predicate**> leader name
          <**object**> Jacob Bundsgaard
T= The leader of Aarhus is Jacob Bundsgaard.

Cleaned E2E
D= name[Zizzi], eatType[coffee shop], area[riverside]
Linearized D= <**name**> name=[Zizzi]; <**area**> area=[riverside]
          <**eatType**> eatType=[coffee shop];
T= You can find a coffee shop named Zizzi in the riverside area.

ViGGO
D= request(developer[EA Canada], specifier[favorite])
Linearized D= <**request**> request (<**developer**> developer: [EA Canada],
          <**specifier**> specifier: [favorite] <**request**>)
T= What's your favorite game that EA Canada has made?

LDC2017T10
D= (r / respond−01
  :ARG0 (c / country :wiki ''United_States''
    :name (n / name :op1 ''United''
    :op2 ''States''))
  :ARG1 (d / develop−01
    :mod (t / that))
  :ARG2 (c2 / condemn−01
    :manner (s / swift)))
Linearized D= (respond <:**ARG0**>
  (country <:**name**> (United States))
  <:**ARG1**> (develop <:**mod**> (that))
  <:**ARG2**> (condemn <:**manner**> (swift)))
T= The United States responded to that development with swift condemnation.

Figure 1: Examples from each of the datasets

Figure 2: Data-to-text language model fine-tuning setup

Inputs: The input sequence is the data D concatenated with the text T: $(<data>\{D\}<text>\{T\})$. The special tokens $<data>$ and $<text>$ are appended to GPT-2's original vocabulary; their embeddings are learnt during fine-tuning. In addition, we append to the vocabulary the *MR*-dependent special tokens described above. After tokenization, we get a sequence S of subword tokens, which are encoded to point to vocabulary indices: $S = (<data>, d_1, \ldots d_k, <text>, t_1, \ldots t_m) = (s_0, \ldots s_n)$.

One interesting feature of GPT-2 is its use of Byte-Pair Encoding (BPE) (Sennrich et al., 2016) on bytes instead of unicode characters. Hence, with a modestly-sized subword vocabulary of around 50K, it can encode any input text and score any output sequence, without suffering from unknown tokens. This is beneficial for our task where named entities are common.

GPT-2 additionally expects positional encodings to help capture the input tokens' order. Our core addition to the model is a third type of input: fine-grained state embeddings. These are analogous to the "Segment Embeddings", introduced in BERT (Devlin et al., 2019) to distinguish between sentence pairs in the next sentence prediction task. However, in our case, the state is defined at a more fine-grained level to give the model a hint on the type of the data being handled. The state vector for S is a vector of tokens with size $|S|$, with each token ID indicating the type of s_i. Our strategy is to decide the state based on the special tokens we inserted in the data processing stage. We use the following rule: the state token ID of any token s_i is the ID of the last special token preceding it (i.e. in the range $(s_0 \ldots s_i)$ inclusively).

Training: The input embeddings, positional embeddings, and state embeddings are summed together and fed to the first GPT-2 layer. The last GPT-2 layer output is then normalized using "LayerNorm" (Ba et al., 2016) before passing it to a linear layer added on top. The weights of the latter are tied to the input embeddings. Finally, a softmax is applied to the linear layer's output to generate probability distributions of the output tokens. Our training objective is a language modeling one where we aim to find the set of weights θ that minimize the cross-entropy loss $\ell = \sum_{i=|D|+2}^{|S|} \log P_\theta(s_i | s_0, \ldots s_{i-1})$.

Note that, since our task is to generate text given the data, the cross-entropy loss is computed for the text following the input data. We mask the data component in the loss above and sum the loss from index $|D| + 2$ (i.e., after the $<text>$ token). We use *AdamW* as an optimizer (Loshchilov and Hutter, 2019).

4.2 Semantic Fidelity Classifier

The Semantic Fidelity Classifier (SFC) provides an additional assessment of how accurately the generated text reflects the input data. A text is deemed to possess semantic fidelity if it accurately conveys all of the input data without omitting any nor adding additional data. Our approach draws parallels between this task and natural language inference (NLI) tasks, where the goal is to determine whether a "hypothesis" is true, false, or undetermined given a "premise". Similarly, in semantic fidelity classification, we

aim to determine if the text is "accurate" or contains some "omission", "repetition", "hallucination", or "value errors" given the data. We cast the problem as a sentence-pair classification task for the (*Data, Text*) pairs, using RoBERTa (Liu et al., 2019) as a base encoder. This formulation has been successfully used for NLI problems before (Devlin et al., 2019).

Training Data Generation: The classifier's training data should consist of semantically faithful and semantically incorrect examples. We generate training data for the SFC automatically from the training data of the main D2T task. We define a set of simple dataset-independent transformations that account for common errors in data-to-text generation. For each tuple (D_i, T_i) in the training data, we split the text T_i into sentences, using the spaCy sentence tokenizer (Honnibal and Montani, 2017). We then generate a set of new tuples for the SFC consisting of (D_i, T_j, l) for each of the labels l below, generated as follows:

- **Accurate**: This is the text T_i.

- **Omission**: Remove the shortest sentence in T_i (to help detect subtle omissions).

- **Repetition**: Take a random sentence in T_i and insert it before another random sentence in T_i.

- **Hallucination**: Select a random sentence from another training text $T_{j \neq i}$ and insert it before a random sentence in T_i.

- **Value Errors**: Select a random value x that occurs verbatim in both D_i and T_i, and replace x in T_i with a random other value from D_i. For slot-based *MR* (Cleaned E2E and ViGGO), x is selected from the slots' values. For graph-based *MR* (LDC2017T10), x is selected from the graph's leaves. For RDF triples (WebNLG dataset), x is chosen from the triples' subjects and objects.

A related approach, with a different setup and modeling architecture, has been used before in the context of consistency in abstractive summarization (Kryściński et al., 2019). There, weakly supervised models trained on domain-specific data have been shown to outperform supervised models trained on out-of-domain, human-annotated data.

Model Input: As shown in Figure 3, we concatenate the data and text tokens, adding the special start ($<s>$) and end ($</s>$) tokens used during the training of RoBERTa. In addition to subword token embeddings, we add positional embeddings (representing the token position) and segment embeddings (representing data vs. text types).

Training: The 3 embeddings are summed element-wise to produce the input representation passed to RoBERTa's first encoder layer. Each layer subsequently applies a self-attention followed by a feed-forward network. We take the output hidden layer corresponding to the first token ($<s>$) and pass it through an additional single-layer neural network. The model is trained as a multi-class classifier with a cross-entropy loss as the objective and AdamW as the optimizer.

Figure 3: Semantic fidelity classifier setup

4.3 Decoder

Our decoding algorithm for the D2T-LM is based on beam-search. At each decoding step, items are ranked according to the score $R = \frac{1}{(i-(|D|+2))^{\alpha}} \prod_{|D|+2}^{i} P(s_i | s_0 \dots s_{i-1})$. The score multiplies the conditional probabilities' product with a length normalization factor. Low-scoring candidates are dropped once the number of candidates exceeds the beam size.

Compared to traditional beam search, we do not aggregate probabilities from the start of the sequence, but from the start of the text component (index $|D|+2$). The length normalization is also adjusted to only account for the text component. We do this because we fine-tuned the D2T-LM to generate text given
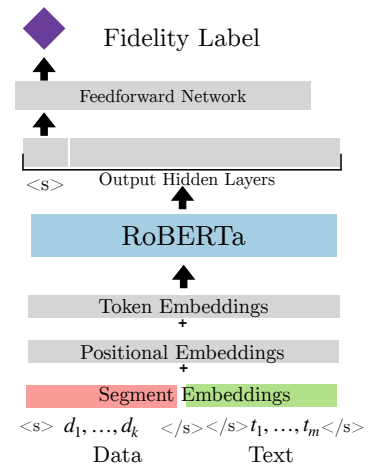
data as context, not to generate the data itself. Hence, we remove the data tokens from the beam-scoring function. In our experiment, we use a value of $\alpha = 0.75$. At the end of the beam-search, we use the SFC to rerank the *complete* candidates (terminated with an end-of-sequence token) in the beam. The reranking metric uses the following binary score: $\mathbb{1}_{SFC(D_i, T_i) = \text{``accurate''}}$.

Hence, we push the text T_i to the top of the beam if our SFC labels the (D_i, T_i) tuple as "*accurate*". We resolve ties using the original D2T-LM scores. An alternative strategy would be to apply reranking at each decoding stage, but we empirically found this strategy to have negligible accuracy gains while requiring a cost that grows with the text size. In addition to helping surface semantically accurate outputs, the SFC labels can be used to assess whether the generated text is usable in practice. In our experiments, we compare this model-based approach to the heuristic approaches commonly used.

5 Experiments

For each dataset, we generate outputs from three versions of DATATUNER for our ablation studies. DATATUNER_NO_FC/FS simply relies on the D2T-LM, with no SFC-based reranking and a coarse-grained version of the state embeddings that contains only *<data>* and *<text>* tokens (as done by Wolf et al. (2019b)). DATATUNER_NO_FC adds the fine-grained state embeddings described in Section 4.1 to DATATUNER_NO_FC/FS. DATATUNER_FC adds the SFC-based reranking. For the SFC, we train the model using the RoBERTa-large model (355M parameters) on lower-cased text. On the synthetic test set generated, the classifier has a macro-averaged F1-score (across 5 classes) of 97%, 97%, 98%, and 98% for the LDC2017T10, WebNLG, Cleaned E2E, and ViGGO datasets respectively. We use the models bundled within the HuggingFace Transformers library (Wolf et al., 2019a). The D2T-LM uses the GPT-2-Medium model (with 345M-parameters) as its base model. The beam search width during decoding is 5. Training was performed on a single machine (Amazon AWS p3.8xlarge). During inference, text from DataTuner is generated at an average rate of 11.8 tokens per second on NVIDIA Tesla K80 GPUs.

We evaluate each variant's outputs with automated metrics and crowdsourced fluency and fidelity evaluation. We also quantify the efficacy of our semantic fidelity classifier with expert-annotations. We compare against the state of the art systems on each dataset, selected based on BLEU scores. These are a graph-optimized Transformer Seq2seq model by Zhu et al. (2019) for LDC2017T10, Transformer-based models (pipeline and E2E) by Castro Ferreira et al. (2019) for WebNLG, an LSTM Seq2seq model with a rule-based reranker by Dušek et al. (2019) for Cleaned E2E, and a Transformer Seq2seq model by Juraska et al. (2019) for ViGGO. The supplementary material contains the outputs from our system variants and the main training hyperparameters (obtained via manual tuning).

5.1 Automated Evaluation

For each test set, we compute BLEU (B) (Papineni et al., 2002), which measures n-gram precision, ME-TEOR (M) (Lavie and Agarwal, 2007), which is based on the harmonic mean of the unigram precision and recall while accounting for stem and synonymy matching, ROUGE_L (R) (Lin, 2004), which calculates recall for the longest common subsequence, and CIDEr (C) (Vedantam et al., 2015), which is based on TF-IDF scoring of n-grams. We used the official evaluation scripts of the E2E challenge[1]. Table 2 compares the results generated by DATATUNER variants against the state of the art.

Improvements from the D2T-LM alone: Comparing the simple DATATUNER_NO_FC/FS model to the state of the art, we find that it already improves the BLEU score across 2 datasets and the METEOR score across 3 datasets. This indicates that the D2T-LM component of DATATUNER is itself contributing to achieving an end-to-end state-of-the-art system that needs no delexicalization or *MR*-specific encoding.

Fine-grained state embeddings matter: Across the 4 datasets, adding fine-grained state embeddings boosts performance on these metrics, with improvements ranging from 0.3 (on Cleaned E2E) to 2.0 BLEU points (on ViGGO).

SFC effect on automated metrics: Several studies highlight shortcomings of automated metrics for evaluating semantic adequacy (Novikova et al., 2017a; Shimorina, 2018). In this vein, compared to our

---

[1] `https://github.com/tuetschek/e2e-metrics`

DATATUNER_NO_FC model, we observe slight additional boosts from introducing the SFC classifier in the DATATUNER_FC variant. Interestingly, DATATUNER_FC always has the highest METEOR score, which was the only metric found by Shimorina (2018) to correlate with semantic adequacy.

Largest boost on the most complex text: DATATUNER had the biggest improvement, 5.9 BLEU points, on the LDC2017T10 dataset. This is interesting given that (1) the text in LDC2017T10 is typically long with more complex sentence structures and that (2) the baseline systems targeting AMR-to-text (Zhu et al., 2019; Guo et al., 2019; Ribeiro et al., 2019) built more sophisticated architectures compared to other datasets (e.g. ViGGO and Cleaned E2E). This illustrates our system's ability to work across a spectrum of data representations and text complexity.

D	Model	B	M	R	C
LDC2017T10	DATATUNER_FC	**37.7**	**38.9**	**65.1**	**3.9**
	DATATUNER_NO_FC	37.2	38.4	65.0	**3.9**
	DATATUNER_NO_FC/FS	35.6	37.3	64.4	3.8
	Zhu et al. (2019)	31.8	36.4	-	-
	Guo et al. (2019)	30.4	-	-	-
	Ribeiro et al. (2019)	27.9	33.2	-	-
WebNLG	DATATUNER_FC	52.4	**42.4**	**66.0**	**3.7**
	DATATUNER_NO_FC	**52.9**	41.9	65.9	**3.7**
	DATATUNER_NO_FC/FS	51.6	40.6	64.9	3.6
	Castro Ferreira et al. (2019) Pipe.	51.7	32.0	-	-
	Castro Ferreira et al. (2019) E2E	33.5	25.0	-	-
	Moryossef et al. (2019b) Pipe.	47.4	39.1	63.1	2.7
Cleaned E2E	DATATUNER_FC	**43.6**	**39.0**	57.5	**2.0**
	DATATUNER_NO_FC	**43.6**	**39.0**	57.5	**2.0**
	DATATUNER_NO_FC/FS	43.3	38.9	**57.6**	**2.0**
	Dušek et al. (2019) (TGen+)	40.5	37.6	56.0	1.8
ViGGO	DATATUNER_FC	**53.6**	**39.4**	**64.0**	**2.7**
	DATATUNER_NO_FC	53.4	39.1	63.8	**2.7**
	DATATUNER_NO_FC/FS	51.4	38.9	62.7	2.5
	Juraska et al. (2019)	52.1	39.1	63.8	2.5

Table 2: Evaluation with automated metrics.

5.2 Human Evaluation of Fluency

We conduct human evaluation of fluency for 150 examples sampled at random from each dataset. We use Amazon's MTurk to ask crowd workers how fluent a text is on a 7-point Likert scale using sliders, where "high fluency" is defined as "grammatical, natural, and could have been produced by a skilled native speaker". Following findings from Novikova et al. (2018) and Van Der Lee et al. (2019) for acquiring more consistent human ratings, texts generated for the same meaning representation by different systems are presented together in a single task for annotators to score them relative to each other. We include the human-written text, and randomize the texts' order. For fair comparison, we lower-case our generated texts for the LDC2017T10 to match the outputs of Zhu et al. (2019). We also detokenize outputs from that work to avoid these biasing the workers. We restrict to US-based annotators who completed >500 tasks, out of which more than 97% had been accepted.

Improvement on the state of the art: As shown in the column "Flu." of Table 3, compared to the human baseline, our DATATUNER_FC model improves the fluency on all four datasets compared to the state of the art systems with statistically significant margins ($p < 0.05$). For computing significance, we use the pairwise Wilcoxon signed-rank test (Wilcoxon, 1992) with the null hypothesis that the fluency values for each pair of systems come from the same distribution. For LDC2017T10, where DATATUNER_FC had the largest gap in BLEU score (+5.9), we observe the widest fluency improvement (+0.82) compared to Zhu et al. (2019). Interestingly, despite the fact that DATATUNER_FC scored 0.7 higher on BLEU compared to the pipeline approach in (Castro Ferreira et al., 2019) for WebNLG, the difference in fluency is 0.69, which is relatively large. We conjecture that this originates from two main sources. First, semantic errors might be perceived by annotators as breaking the fluency. For example, one text contained the phrase "*has a runway length of Shehbaz Sharif*". Second, the pipeline approach had a sizeable portion of non-realized outputs (e.g. "PATIENT-1 is made with PATIENT-1 and PATIENT-2."), which were annotated as non-fluent. On the closed-domain datasets (ViGGO and Cleaned E2E), the fluency margins are smaller while still statistically significant. This is expected as these datasets have a narrow set of sentence formulations that are easier to learn.

Improvement on the human baseline: Surprisingly, we find that DATATUNER_FC received a higher overall average fluency score on 3 datasets compared to the human baseline. This difference is statistically significant in both Cleaned E2E and ViGGO, with the largest difference being 1.04 points in Cleaned E2E. Investigating, we found several low-scored texts had an informal style and problems in sentence construction. One example contained "*It serves Chinese food for less.*" One explanation could be that, once fine-tuned on a large enough dataset, our models have less tendency to deviate from common formulations that annotators prefer.

5.3 Crowdsourced Evaluation of Fidelity

We take a two-step approach for evaluating semantic fidelity with both crowdworkers and expert annotators. We start with a crowdsourcing experiment involving the same 150 randomly sampled examples from each dataset used in the fluency evaluation. We use Amazon's MTurk (with the same restrictions on annotators) and present texts for the same *MR* together. To avoid requiring non-expert annotators to understand the *MR*, we present a human reference text against which to compare the system outputs. We ask annotators to make a choice whether each text is "accurate" or "inaccurate", i.e. whether it "conveys all the factual information in the original text" without any "information missing, added or repeated". We ask annotators to ignore grammar quality or style differences, provided that the overall meaning is the same. Three annotators com-

D	Model	Flu.	Fid.	DSA	$\mathbf{Q}_D$	HSA	$\mathbf{Q}_H$
LDC2017T10	DATATUNER_FC	$4.79^{s,h}$	78.0^s	$81.8^{s,h}$		—	
	DATATUNER_NO_FC	4.87^s	77.3^s	$70.5^{s,h}$		—	
	DATATUNER_NO_FC/FS	$4.78^{s,h}$	68.7	$65.8^{s,h}$	90.8	—	-
	Zhu et al. (2019)	3.97^h	58.7	62.4^h		—	
	Human	5.05	—	93.1		—	
WebNLG	DATATUNER_FC	5.23^s	84.0^s	$91.7^{s,h}$		$58.1^{s,h}$	
	DATATUNER_NO_FC	5.20^s	80.0^s	$81.4^{s,h}$		$54.1^{s,h}$	
	DATATUNER_NO_FC/FS	5.27^s	72.0^s	$73.6^{s,h}$	**87.5**	$43.6^{s,h}$	73.3
	Castro Ferreira et al. (2019)	4.54^h	44.0	50.5^h		33.7^h	
	Human	5.21	—	94.7		41.2	
Cleaned E2E	DATATUNER_FC	$5.46^{s,h}$	92.0^s	$99.3^{s,h}$		$97.3^{s,h}$	
	DATATUNER_NO_FC	$5.46^{s,h}$	90.7^s	99.0^h		$97.1^{s,h}$	
	DATATUNER_NO_FC/FS	$5.45^{s,h}$	88.7	98.9^h	**89.2**	$97.1^{s,h}$	75.0
	Dušek et al. (2019) TGen+	5.23^h	86.0	98.9^h		98.0^h	
	Human	4.42	—	99.9		100.0	
ViGGO	DATATUNER_FC	$5.77^{s,h}$	86.0	97.2^s		$74.5^{s,h}$	
	DATATUNER_NO_FC	$5.76^{s,h}$	82.7	92.8^h		$82.3^{s,h}$	
	DATATUNER_NO_FC/FS	5.60^h	81.3	91.7^h	92.5	$82.5^{s,h}$	88.3
	Juraska et al. (2019)	5.58^h	80.7	92.8^h		90.9	
	Human	5.34	—	97.1		91.9	

Table 3: Crowdsourced evaluation of fluency (Flu.), crowdsourced evaluation of semantic fidelity (Fid.), DATATUNER Semantic Accuracy (DSA), Heuristic Semantic Accuracy (HSA), and quality measures $\mathbf{Q}_D$ and $\mathbf{Q}_H$ for DSA and HSA. Superscripts s and h indicate a statistically significant difference vs. the state of the art and the human baseline respectively. Bolded Q_D values are higher with statistical significance when compared to the corresponding Q_H values.

plete each task, and we take the mode result for each text, assuming "inaccurate" in the event of a tie.

DATATUNER has higher semantic accuracy in human evaluation The results presented in the "Fid." column of Table 3 show that DATATUNER_FC has a superior accuracy to all the other variants as well as the state of the art models. The differences to the state of the art models range from 5.3% (ViGGO) to 40% (WebNLG) and are statistically significant on WebNLG, LDC2017T10, and Cleaned E2E ($p <$ 0.05, as measured by McNemar's test (McNemar, 1947)). The trend among the DATATUNER variants also shows a clear impact of both the fine-grained state embeddings and the SFC on boosting the overall accuracy of the generated text.

5.4 Expert Evaluation of Fidelity

Next, we assess the semantic fidelity with experts' annotations. We have two goals in this section: (1) comparing our model-based approach to heuristic-based approaches as automated methods of judging semantic accuracy, and (2) using this comparison outcome to illustrate that DATATUNER delivers higher semantic accuracy as measured by the better fidelity metric.

The baseline method uses heuristics to label each data-text tuple as accurate (A_H) or erroneous (E_H). For this, we use the heuristics by Shimorina and Gardent (2018) for WebNLG, by Juraska et al. (2018) for ViGGO, and by Dušek et al. (2019) for Cleaned E2E. We are not aware of heuristic-based scripts for LDC2017T10. We compute Heuristic Semantic Accuracy (HSA) of a dataset as the fraction with the label A_H. Our method uses the SFC component in DATATUNER to assign accurate (A_D) or erroneous (E_D) labels to each data-text tuple. We compute DATATUNER Semantic Accuracy (DSA) as the fraction with the label A_D. Both metrics are computed per system across each dataset.

To compare the quality of HSA and DSA as measures of semantic accuracy, we manually annotated a sample of data-text tuples. Since the vast majority of texts are expected to be accurate, especially on the cleaner datasets, we designed a sampling methodology to give a balanced representation of semantically accurate and inaccurate texts. To start, we sample 4 indices from the target dataset such that the human baseline outputs for these indices are labeled as: $\{(A_H, E_D), (E_H, A_D), (E_H, E_D), (A_H, A_D)\}$. We do the same with the state of art system and DATATUNER_FC outputs. We continue in a round-robin fashion until we get 24 indices per dataset. For LDC2017T10 dataset, we sample 24 indices in a similar fashion while ignoring the A_H and E_H labels. Next, two authors were presented with the input meaning representation and the output texts generated by each system (in a random order) for the 24 sampled entries. The authors manually labeled the resulting 480 data-text tuples as accurate (A_M) or erroneous (E_M). Inter-annotator agreement measured with Cohen's Kappa was 0.81, indicating near-perfect agreement. We use these labels to assess the quality Q_D of the DSA metric as the percentage of cases where the manual label A_M matches A_D. Similarly, we evaluate the quality Q_H of the HSA metric as the percentage of cases where A_M matches A_H. These percentages are aggregated across systems, obtaining 120 samples per dataset. We present these metrics in Table 3.

DSA provides higher quality semantic evaluation: We notice that Q_D is 4.2% higher on ViGGO and 14.2% higher on both Cleaned E2E and WebNLG, compared to Q_H. These differences are statistically significant ($p < 0.05$) on WebNLG and Cleaned E2E, as measured by McNemar's test (McNemar, 1947) with the null hypothesis that the marginal probability for each outcome (*accurate* or *erroneous*) is the same for both algorithms.

HSA struggles with open domains: The heuristic-based approach labeled only 41.2% of the human references in WebNLG as accurate, 16.9% lower than the score it assigned to our DATATUNER_FC. Since the latter was trained on human references, this difference is more likely to stem from shortcomings of this approach for assessing the semantics. Checking the data, we observed that humans tend to create more diverse formulations, such as converting *United Kingdom* to *UK*, which are easy to miss with heuristics. On the contrary, our DSA metric scored the human references higher.

DATATUNER_FC delivers higher semantic accuracy: Now that we have established that DSA is a better measure of semantic accuracy compared to HSA, we can see from Table 3 that, across all datasets, DATATUNER_FC significantly improves over the state of the art models as measured by the DSA metric (McNemar's test gives $p < 0.05$). Compared to other DATATUNER variants, DATATUNER_FC adds between 0.3% and 11.3% improvements, corroborating the utility of the semantic fidelity classifier. Finally, we note that, since the baseline models for Cleaned E2E and ViGGO use the heuristics for reranking their outputs, they are expected to show higher HSA. However, what our manual annotations prove is that the HSA metric itself is of lower quality compared to the DSA metric.

6 Conclusion

We presented DATATUNER, an end-to-end data-to-text generation system equipped with a semantic fidelity classifier. DATATUNER records new state of the art results on four different datasets, with significant margins on automated metrics. We also show that our system has a clear fluency advantage over all the previous state of the art models. We further illustrate DATATUNER's strengths for delivering semantically accurate outputs.

References

Shubham Agarwal, Marc Dymetman, and Éric Gaussier. 2018. Char2char generation with reranking for the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 451–456, Tilburg University, The Netherlands, November. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 171–176, Tilburg University, The Netherlands, November. 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)*, pages 552–562, Hong Kong, China, November. Association for Computational Linguistics.

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

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-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 3649–3658, Minneapolis, Minnesota, June. Association for Computational 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic Noise Matters for Neural Natural Language Generation. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188. 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. *Journal of Artificial Intelligence Research*, 61:65–170.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312, March.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162, New Orleans, Louisiana, June. Association for Computational Linguistics.

Juraj Juraska, Kevin K Bowden, and Marilyn Walker. 2019. Viggo: A video game corpus for data-to-text generation in open-domain conversation. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 584–593, Tokyo, Japan, October–November. Association for Computational Linguistics.

Kevin Knight, Laura Baranescu Bianca Badarau, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2017. Abstract meaning representation (AMR) annotation release 2.0 LDC2017T10. *Web Download. Philadelphia: Linguistic Data Consortium.*

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840.*

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas, November. Association for Computational Linguistics.

Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore, August. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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. *arXiv preprint arXiv:1907.11692.*

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Fabrizio Morbini, Kartik Audhkhasi, Ron Artstein, Maarten Van Segbroeck, Kenji Sagae, Panayiotis Georgiou, David R Traum, and Shri Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 49–54. IEEE.

Amit Moryossef, Ido Dagan, and Yoav Goldberg. 2019a. Improving quality and efficiency in plan-based neural data-to-text generation. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019b. 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, June. Association for Computational Linguistics.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. 2018. Operation-guided neural networks for high fidelity data-to-text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3879–3889, Brussels, Belgium, October-November. Association for Computational Linguistics.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy, July. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017a. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark, September. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 72–78, New Orleans, Louisiana.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Steffen Pauws, Albert Gatt, Emiel Krahmer, and Ehud Reiter. 2019. Making effective use of healthcare data using data-to-text technology. In *Data Science for Healthcare*, pages 119–145. Springer.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Leonardo FR Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing amr-to-text generation with dual graph representations. 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)*, November.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative 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 4060–4067, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. *INLG 2018*, page 360.

Anastasia Shimorina. 2018. Human vs automatic metrics: on the importance of correlation design. *arXiv preprint arXiv:1805.11474*.

Charese Smiley, Frank Schilder, Vassilis Plachouras, and Jochen L. Leidner. 2017. Say the right thing right: Ethics issues in natural language generation systems. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 103–108, Valencia, Spain, April. Association for Computational Linguistics.

Chris Van Der Lee, Albert Gatt, Emiel Van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG'19)*, Tokyo, Japan. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*.

Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

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

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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)*, pages 5458–5467, Hong Kong, China, November. Association for Computational Linguistics.

Appendix A. Examples of Generated Text

In Table 4, we present examples of the outputs generated by DATATUNER and the state of the art models, alongside the human references.

D	Model	Examples
Cleaned E2E	Human	pub in riverside that also serves Italian food: The Vaults has got high prices, is not child-friendly and has an average rating. It is near Rainbow Vegetarian Café in riverside.
	DATATUNER_FC	The Vaults is an Italian pub in the riverside area near Rainbow Vegetarian Café. It has an average customer rating and a high price range. It is not child friendly.
	Dušek et al. (2019)	The Vaults is an italian pub with an average customer rating. It is located in the riverside area near Rainbow Vegetarian Café. It is not children friendly and has a high price range.
WebNLG	Human	Adirondack regional airport serves both Lake Placid and Saranac Lake, New York. The length of the runway at Asirondack regional airport is 2003.
	DATATUNER_FC	Adirondack Regional Airport serves the cities of Lake Placid and Saranac Lake, New York and has a runway length of 2003.0.
	Castro Ferreira et al. (2019)	Adirondack Regional Airport serves the city of Lake Placid and Saranac Lake, New York and has a runway length of Shehbaz Sharif.
LDC2017T10	Human	the plan requires 8 precautionary steps before the order to shoot down a plane may be issued.
	DATATUNER_FC	the plan requires eight precautionary steps before the order to shoot down the plane can be issued.
	Zhu et al. (2019)	the plan required 8 precaution steps before it can be issued to order shot down.
ViGGO	Human	Guitar Hero: Smash Hits was a very bad game. 2009 was a terrible year for gaming and I just can't stand the games released that year.
	DATATUNER_FC	Guitar Hero: Smash Hits is a really bad game. 2009 was a really bad year for games
	Juraska et al. (2019)	Guitar Hero: Smash Hits is a very bad game, especially because it came out in 2009.

Table 4: Examples of text generated by the different models