

Sequence-to-Sequence Models for Data-to-Text Natural Language Generation: Word- vs. Character-based Processing and Output Diversity

Glorianna Jagfeld, Sabrina Jenne, Ngoc Thang Vu

Institute for Natural Language Processing (IMS)

Universität Stuttgart, Germany

{jagfelga, beersa, thangvu}@ims.uni-stuttgart.de

Abstract

We present a comparison of word-based and character-based sequence-to-sequence models for data-to-text natural language generation, which generate natural language descriptions for structured inputs.

On the datasets of two recent generation challenges, our models achieve comparable or better automatic evaluation results than the best challenge submissions. Subsequent detailed statistical and human analyses shed light on the differences between the two input representations and the diversity of the generated texts. In a controlled experiment with synthetic training data generated from templates, we demonstrate the ability of neural models to learn novel combinations of the templates and thereby generalize beyond the linguistic structures they were trained on.

1 Introduction

Natural language generation (NLG) is an actively researched task, which according to [Gatt and Krahmer \(2018\)](#) can be divided into text-to-text generation, such as machine translation ([Koehn, 2017](#)), text summarization ([See et al., 2017](#)), or open-domain conversation response generation ([Vinyals and Le, 2015](#)) on the one hand, and data-to-text generation on the other hand. Here, we focus on the latter, the task of generating textual descriptions for structured data. Data-to-text generation comprises the generation of system responses based on dialog acts in task-oriented dialog systems ([Wen et al., 2015b](#)), sport games reports and weather forecasts ([Angeli et al., 2010](#)), and database entry descriptions ([Gardent et al., 2017a](#)). In this paper, we focus on sentence planning and surface realization. We build on data-to-

text datasets of two recent shared tasks for end-to-end NLG, namely the E2E challenge ([Novikova et al., 2017b](#)) and WebNLG challenge ([Gardent et al., 2017b](#)). Example input-text pairs for both datasets are shown in Figure 1.

Neural sequence to sequence (Seq2Seq) models ([Graves, 2013](#); [Sutskever et al., 2014](#)) have shown promising results for this task, especially in combination with an attention mechanism ([Bahdanau et al., 2014](#); [Luong et al., 2015](#)). Several recent NLG approaches ([Dušek and Jurčicek, 2016](#); [Mei et al., 2016](#); [Kiddon et al., 2016](#); [Agarwal and Dymetman, 2017](#)), as well as most systems in the E2E and WebNLG challenge are based on this architecture. While most NLG models generate text word by word, promising results were also obtained by encoding the input and generating the output text character-by-character ([Lipton et al., 2015](#); [Goyal et al., 2016](#); [Agarwal and Dymetman, 2017](#)). Five out of 62 E2E challenge submissions operate on the character-level. However, it is difficult to draw conclusions from the challenge results with respect to this difference, since the submitted systems also differ in other aspects and were evaluated on a single dataset only.

Besides adequacy and fluency, variation is an important aspect in NLG ([Stent et al., 2005](#)). In addition to comparing the linguistic and content-wise correctness of word- and character-based Seq2Seq models through automatic and human evaluation, we investigate the variety of their outputs. While template-based systems can assure perfect content and linguistic quality, they often suffer from low diversity. Conversely, neural models might generalize beyond a limited amount of training texts or templates, thereby producing more diverse outputs. To test this hypothesis, we train Seq2Seq models on template-generated texts with a controlled amount of variation and show that they not only reproduce the templates, but also

E2E input: name[Midsummer House], customer rating [average], near [The Bakers]
reference 1: Customers gave Midsummer House, near The Bakers, a 3 out of 5 rating.
reference 2: Midsummer house has an average customer rating and is near The Bakers.
delexicalized input: name[NAME], customer rating [average], near [NEAR]
delexicalized reference 1: Customers gave NAME, near NEAR, a 3 out of 5 rating.

WebNLG input: cityServed(Abilene Regional Airport[Abilene]), isPartOf(Abilene[Texas])
reference 1: Abilene is in Texas and is served by the Abilene regional airport.
reference 2: Abilene, part of Texas, is served by the Abilene regional airport.
delexicalized input: city served(AGENT-1[BRIDGE-1]), is part of(BRIDGE-1[PATIENT-1])
delexicalized reference 1: BRIDGE-1 is in PATIENT-1 and is served by the AGENT-1.

Figure 1: Example input-reference pairs from the E2E and WebNLG development set.

generate novel structures resulting from template combinations.

In sum, we make the following contribution:

- We compare word- and character-based Seq2Seq models for NLG on two datasets.
- We conduct an extensive automatic and manual analysis of the generated texts and compare them to human performance.
- In an experiment with synthetic training data generated from templates, we demonstrate the ability of neural NLG models to learn template combinations and thereby generalize beyond the linguistic structures they were trained on.

2 Related Work

This section reviews relevant related work according to the two main aspects of this paper: different input and output representations for data-to-text NLG as well as measuring and controlling the variation in the generated outputs.

2.1 Input and Output Representations

While the first NLG systems relied on hand-written rules or templates that were filled with the input information (Cheyer and Guzzoni, 2006; Mirkovic et al., 2006), the availability of larger datasets has accelerated the progress in statistical methods to train NLG systems from data-text pairs in the last twenty years (Oh and Rudnicky, 2000; Mairesse and Young, 2014). Generating output via language models based on recurrent neural networks (RNNs) conditioned on the input (Sutskever et al., 2011) proved to be an effective method for end-to-end NLG (Wen et al., 2015a,b, 2016).

The input can be represented in several ways:

(1) In a discrete vector space via one-hot-vectors (Wen et al., 2015a,b), or in a continuous space either (2) by encoding fixed-size input information in a feed-forward neural network (Zhou et al., 2017; Wiseman et al., 2017) or (3) by the means of an encoder RNN, which processes variable-sized inputs sequentially, giving rise to the Seq2Seq architecture.

Character-based Seq2Seq models were first proposed for neural machine translation (Ling et al., 2015; Chung et al., 2016; Lee et al., 2017). Their main advantage over word-based models is that they can represent an unlimited word inventory with a small vocabulary. They can learn to copy any string from the input to the output, which is especially useful for data-to-text NLG, as information from the input such as the name of a restaurant or a database entity is often expected to appear verbatim in the generated text. Word-based models, in contrast, have to make use of delexicalization during pre- and postprocessing (Wen et al., 2015b; Dušek and Jurčicek, 2016) or have to apply dedicated copy mechanisms (Gu et al., 2016; See et al., 2017; Wiseman et al., 2017) to handle open vocabularies. The other side of the coin is that sequences are much longer in character-based processing, implying longer dependencies and more computation steps for encoding and decoding.

Subword-based representations (Sennrich et al., 2016; Wu et al., 2016) can offer a trade-off between word- and character-based processing and are a popular choice in NMT and summarization (See et al., 2017). Here, the vocabulary consists of subword units of different lengths, which are assigned by minimizing the entropy on the training set. We also experimented with such representations in preliminary experiments, but

found them to perform much worse than word- or character-based representations. Our impression is that recurring entity names in the training data coming from multiple reference texts for the same input lead to overfitting on the training vocabulary and to poor generalization to novel inputs. This is also reflected by the rather unsatisfying performance of subword-based approaches in the E2E¹ and WebNLG challenge (ADAPT system (Gardent et al., 2017b)).

2.2 Output Diversity

Evaluation of data-to-text NLG has traditionally centered around semantic fidelity, grammaticality, and naturalness (Gatt and Krahmer, 2018; Oraby et al., 2018b). More recently, the controllability of the style of the outputs and their variation has moved into focus as well (Ficler and Goldberg, 2017; Herzig et al., 2017; Oraby et al., 2018b,a).

Oraby et al. (2018b) showed that the n-gram entropy of the outputs of a neural NLG system is significantly lower compared to its training data. This can be seen as evidence that the NLG system extracts only a few dominant patterns from the training data that it will generate over and over. Without explicit supervision signals, neural NLG models cannot distinguish linguistic or stylistic variation from noise. In the context of image caption generation Devlin et al. (2015) found Seq2Seq models to exactly reproduce sentences from their training data for 60% of the test instances.

Several approaches have been proposed to control NLG outputs with respect to certain stylistic aspects, e.g., mimicking a specific persona or character (Lin and Walker, 2011; Walker et al., 2011; Li et al., 2016), personality traits (Mairesse and Walker, 2008; Herzig et al., 2017; Oraby et al., 2018b,a), or various linguistic aspects such as formality, voice, descriptiveness (Ficler and Goldberg, 2017; Bawden, 2017; Niu et al., 2017). All share the feature that the NLG model is conditioned on a representation of the desired aspect in addition to the usual semantic input representation. While this approach makes it possible to successfully control particular, clearly defined aspects of the generated texts, further research is needed to grant more flexible and comprehensive NLG output control.

¹The subword-based *bzhang_submit* system has the second best ROUGE-L score, but ranks poorly in terms of BLEU and quality in the human evaluation, see <http://www.macs.hw.ac.uk/InteractionLab/E2E/#results>.

3 Models

To encode variable-length inputs and generate variable-length texts, we implement a standard Seq2Seq model (Cho et al., 2014) with Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) and attention. Given a training dataset of input-text pairs $D = \{(x^1, y^1), (x^2, y^2) \dots\}$, the model encodes an input sequence $x = \{x_1 \dots x_n\}$ of symbols x_i into a sequence of hidden states $\{h_1 \dots h_n\}$ by applying a recurrent neural network (RNN) with LSTM cells that can store and forget sequence information:

$$h_t = \text{LSTM}(X^{\text{in}}x_t, h_{t-1}) \quad (1)$$

The decoder generates the output sequence $y_1 \dots y_m$ one symbol y_t at a time by computing $p(y_t | y_1 \dots y_{t-1}, x) = \text{softmax}(W^{\text{out}}(c_t))$.

The decoder output c_t , also referred to as context vector, summarizes the input information in each decoding step as weighted sum of the encoder hidden states: $c_t = \sum_{i=1}^n \alpha_{ti} h_i$. The attention weights α_{ti} are computed with the general attention mechanism $\alpha_{ti} = \text{softmax}(s_t W^a h_i)$ (Luong et al., 2015). The decoder hidden states s_t are computed recursively based on the previous output token and decoder output:

$$s_t = \text{LSTM}((X^{\text{out}}(y_{t-1}) \circ c_{t-1}), s_{t-1}) \quad (2)$$

s_0 is initialized to the final encoder hidden state h_n , h_0, c_{-1} are initialized to 0; \circ denotes concatenation. The parameters of the models are the input and output embedding matrices $X^{\text{in}}, X^{\text{out}}$, the encoder and decoder LSTM parameters, the attention matrix W^a and the output matrix W^{out} . They are optimized by minimizing the cross entropy of the generated texts y^j with the given references \bar{y}^j for each example in the training set.

Instead of forcing the decoder to decide on a single output symbol in each decoding step, we apply beam search (Cho et al., 2014; Bahdanau et al., 2014) to explore n -best partial hypotheses in parallel.

In the word-based model, each input symbol x_t and output symbol y_t denotes a token. In contrast, in the character-based model, each input and output symbol denotes a single character. Our models learn separate encoder and decoder embedding matrices.

4 Data

We use two recently collected crowd-sourced data-to-text datasets since they are larger and offer more linguistic variety than previously available datasets (Novikova et al., 2017b; Gardent et al., 2017a). The E2E dataset (Novikova et al., 2017b) consists of 47K restaurant descriptions based on 5.7K distinct inputs of 3-8 attributes (*name*, *area*, *near*, *eat type*, *food*, *price range*, *family friendly*, *rating*), split into 4862 inputs for training, 547 for development and 630 for testing. The WebNLG dataset (Gardent et al., 2017a) contains 25K verbalizations of 9.6K inputs composed of 1-7 DBpedia triples from 15 categories such as *athletes*, *comic characters*, *food*, *sport teams*. It is divided into 6893 inputs for training, 872 for development and 1862 for testing. Both datasets have multiple verbalizations for each input. On average there are 8.3 (min. 1, max. 46) verbalizations per input in the E2E dataset and 2.63 (min. 1, max. 12) in the WebNLG dataset, respectively.

To preprocess both datasets, we lowercase all inputs and references and represent the inputs in the bracketed format as shown in Figure 1. For the word-based processing we additionally tokenize the texts with the nltk-tokenizer (Bird et al., 2009) and apply delexicalization, as also illustrated in Figure 1. For the E2E dataset we adopt the challenge’s baseline delexicalization strategy (Dušek and Jurčicek, 2016), which replaces the values of the two open-class attributes *name* and *near* in the input and references by placeholders. For the WebNLG dataset, we adopt the delexicalization strategy of the TILBURG submissions to the challenge, since it performed well and does not require external information. They replaced the subject and object entities of the DBpedia triples in the input and text by numbered placeholders AGENT-N, PATIENT-N, BRIDGE-N, depending on whether they only appear as subject, object or in both roles in the input of an instance. Additionally, we split properties at the camel case in this dataset for both the word- and character-based models as proposed by the ADAPT and MELBOURNE submissions. Table 1 displays statistics for both datasets and processing types.

5 Experiments

We conduct our experiments with the OpenNMT toolkit (Klein et al., 2017), which we extend to also perform character-based processing. We

	E2E		WebNLG	
	word	char.	word	char.
avg. input length	28.5	106.0	24.8	139.8
avg. text length	20.0	109.3	18.8	117.1
input vocabulary	48	39	312	78
output vocabulary	2,721	53	4,264	83

Table 1: E2E and WebNLG training split statistics for word-based processing after delexicalization and character-based processing.

tuned the hyperparameters for each dataset and processing method to optimize the BLEU score on the development sets. The word-based model for the E2E dataset is trained by stochastic gradient descent (SGD) (Robbins and Monro, 1951) and an initial learning rate of 1.0. For all other models, we achieved better performance with the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.001. If there is no improvement in the development perplexity, or in any case after the eighth epoch, we halve the learning rate. Also, we clip all gradients to a maximum of five. We use a batch size of 64. To prevent overfitting, we drop out units in the context vectors with a probability of 0.3. We keep the model with the lowest development perplexity in 13 training epochs.

The word-based E2E model has 64-dimensional word embeddings and a single encoder and decoder layer with 64 units each. All other models use 500-dimensional word- or character embeddings and two layers in the encoder and decoder with 500 dimensions each. While a unidirectional encoder was sufficient for the word-based models, bidirectional encoders were beneficial for the character-based models on both datasets.

We use a beam size of 15 for decoding with the word-based models, and found a smaller beam of five to yield better results for the character-based models. This is probably due to the much smaller vocabulary size of the character-based models.

For automatic evaluation, we report BLEU (Papineni et al., 2002), which measures the precision of the generated n-grams compared to the references, and recall-oriented ROUGE-L (Lin, 2004), which measures the longest common subsequence between the generated texts and the references. We compute these scores with the E2E challenge evaluation script².

²<https://github.com/tuetschek/e2e-metrics>

6 Results and Analysis

Table 2 and 3 display the results on the E2E and WebNLG test sets for models of the respective challenges and our own models³. Since the performance of neural models can vary considerably due to random parameter initialization and randomized training procedures (Reimers and Gurevych, 2017), we train ten models with different random seeds for each setting and report the average (avg) and standard deviation (SD).

On the E2E test set, our single best word- and character-based models reach comparable results to the best challenge submissions. The word-based models achieve significantly higher BLEU and ROUGE-L scores than the character-based models⁴. On the WebNLG test set, the BLEU score of our best word-based model outperforms the best challenge submission by a small margin. The character-based model achieves a significantly higher ROUGE-L score than the word-based model, whereas the BLEU score difference is not significant. In the following, we analyze our models in more detail.

6.1 Analysis of Within-Model Performance Differences

The large performance span of the character-based models on the E2E dataset is due to a single outlier model; the second worst model scores 64.5 BLEU points. The worst-scoring model had a lower accuracy of 91.8% on the development set, whereas all other models scored above 92.2%. To gain more insight on what might constitute the large performance difference, we manually compared the generated texts for ten randomly selected inputs for each number of attributes (60 inputs in total) of the character-based model with the best and worst BLEU score. We found that the worst model makes many mistakes on inputs with three to five attributes, often adding, modifying or removing information, whereas the outputs are mostly correct for inputs with six attributes or more. For these, the outputs of the model with the lowest BLEU score are occasionally even better than those of the best model, which often omits in-

³For an exact comparison, we recomputed the WebNLG challenge results with the E2E evaluation script. They are usually 1-2 points below the scores reported by Gardent et al. (2017b).

⁴All tests for significance in this paper are conducted with Wilcoxon rank sum tests with Bonferroni correction at a p-level of 0.05.

formation (mainly concerning the attribute *family friendly*). We conclude that the large performance difference might be caused by automatic evaluation measures punishing additions more severely than omissions.

We also observe a large performance span for the WebNLG word-based models. Here, we have two models that score exceptionally well with 57.4/58.4 BLEU points, whereas the remaining eight models only obtain BLEU scores in a range of 43.8-48.1. Again, we observe that better models in terms of BLEU score obtain higher accuracies on the development set. We manually compared the outputs of ten randomly chosen inputs for each number of input triples (75 inputs in total) for the model with the highest and lowest BLEU score. In this case, we found that the large difference in the automatic evaluation measures seems justified: The low-scoring model often hallucinates information not present in the input and generally produces many ungrammatical texts, which is not the case for the best model.

system	BLEU	ROUGE-L
challenge		
baseline	65.9	68.5
Thomson Reuters (np 3)	68.1	69.3
Thomson Reuters (np 4)	67.4	69.8
HarvardNLP & H. Elder	67.4	70.8
own		
word	67.8±0.8	70.4±0.6
character	64.6±6.0	67.9±4.7
word (best on dev.)	67.8	70.2
char. (best on dev.)	67.6	70.4

Table 2: E2E test set results. Own results correspond to avg±SD of ten runs and single result of best models on the development set.

6.2 Automatic Evaluation of Human Texts

To gain an impression of the expressiveness of the automatic evaluation scores for NLG, we computed the average scores that the human references would obtain. Table 4 shows the BLEU and ROUGE-L development set scores when treating each human reference as prediction once and evaluating it against the remaining references, compared to the scores of the word-based and

system	BLEU	ROUGE-L
challenge		
baseline	32.1	43.3
MELBOURNE	43.4	61.0
TILBURG-SMT	43.1	58.0
UPF-FORGE	37.5	58.8
own		
word (best on dev.)	44.2	60.9
char. (best on dev.)	41.3	58.4
word	37.0 \pm 3.8	56.3 \pm 2.6
character	39.7 \pm 1.7	58.4 \pm 0.7

Table 3: WebNLG test set results. Own results correspond to single best model on development set and avg \pm SD of ten runs.

character-based models⁵. Strikingly, on the E2E development set, both model variants significantly outperform human texts by far with respect to both automatic evaluation measures. While the human BLEU score is significantly higher than those of both systems on the WebNLG development set, there is no statistical difference between human and system ROUGE-L scores. This further demonstrates the limited utility of BLEU and ROUGE-L scores to evaluate NLG outputs, which was previously suggested by weak correlations of such scores with human judgments (Scott and Moore, 2006; Reiter and Belz, 2009; Novikova et al., 2017a). Furthermore, the high scores on the E2E dataset imply that the models succeed in picking up patterns from the training data that transfer well to the similar development set, whereas human variation and creativity are punished by lexical overlap-based automatic evaluation scores.

6.3 Manual Error Analysis

Since the expressiveness of automatic evaluation measures for NLG is limited, as shown in the previous subsection, we performed a manual error analysis on inputs of each length. We define the input length as the number of input attributes for the E2E dataset, ranging from three to eight, and number of input triples for the WebNLG dataset, ranging from one to seven. We randomly selected

⁵For a fair comparison between human and model performance, we randomly removed one reference for each instance in the models’ evaluation to ensure the same average number of references. We excluded 55 WebNLG instances that had only one reference.

metric	human	word	char.
E2E			
BLEU	55.5 \pm 0.7	68.2 \pm 1.4	65.8 \pm 2.6
ROUGE-L	62.0 \pm 0.4	72.1 \pm 0.7	69.8 \pm 2.6
WebNLG			
BLEU	48.3 \pm 0.7	40.6 \pm 4.2	43.7 \pm 2.4
ROUGE-L	62.4 \pm 0.3	58.5 \pm 3.0	63.1 \pm 0.8

Table 4: E2E and WebNLG development set results in the format avg \pm SD. Human results are averaged over using each human reference as prediction once.

	E2E		WebNLG	
	word	char.	word	char.
content errors				
info. dropped	40.0	30.0	42.9	66.7
info. added	0.0	0.0	6.7	1.9
info. modified	4.4	0.0	19.0	1.9
info. repeated	0.0	0.0	15.2	28.6
linguistic errors				
punctuation errors	5.6	5.6	8.6	3.8
grammatical errors	13.3	14.4	15.2	12.4
spelling mistakes	0.0	0.0	9.5	5.7
overall correctness				
content correct	55.6	70.0	46.7	31.4
language correct	83.3	81.1	69.5	79.0
all correct	48.9	61.1	33.3	26.7

Table 5: Percentage of affected instances in manual error analysis of 15 randomly selected development set instances for each input length.

15 development instances for each input length, resulting in a total of 90 annotated E2E instances and 105 WebNLG instances.

One annotator (one of the authors of this paper) manually assessed the outputs of the models that obtained the best development set BLEU score as summarized in Table 5⁶. As we can see from the bottom part of the table, all models struggle more with getting the content right than with producing linguistically correct texts; 70-80% of the texts generated by all models are completely correct linguistically.

⁶Although multiple annotators could increase the reliability of these results, the annotator reported that the task was very straightforward. We do not expect marking content and linguistic errors to lead to annotator disagreements, with the exception of accidentally missed errors.

	E2E			WebNLG		
	human	word	character	human	word	character
unique sents.	866.3±16.5	203.5±30.6	366.8±60.0	1,185.0±12.6	603.7±144.3	875.4±30.2
unique words	419.7±16.7	64.4±2.3	73.1±7.2	1447.3±7.4	620.3±35.5	881.5±26.0
word E	6.5±0.0	5.1±0.0	5.5±0.0	7.1±0.0	6.3±0.0	6.6±0.0
1-3-grams E	10.4±0.0	7.7±0.1	8.2±0.1	11.6±0.0	10.1±0.1	10.5±0.1
% new texts	99.7±0.2	98.2±0.3	98.8±0.2	91.1±0.3	69.8±4.8	87.5±0.6
% new sents.	85.1±1.1	61.8±6.4	71.4±4.7	87.4±0.4	57.2±5.8	82.1±1.2

Table 6: Linguistic diversity of development set references and generated texts as avg±SD. ‘% new’ denotes the share of generated texts or sentences that do not appear in training references. Higher indicates more diversity for all measures.

Comparing the two datasets, we again observe that the WebNLG dataset is much more challenging than the E2E dataset, especially with respect to correctly verbalizing the content. This can be attributed to the increased diversity of the inputs and texts and to the limited availability of training data for this dataset (cf. Table 1). Moreover, spelling mistakes only appeared in WebNLG texts, mainly concerning omissions of accents or umlauts. This also indicates that there is too few and noisy data for the models to learn the correct spelling of all words. Notably, we did not observe any non-words generated by the character-based models.

The most frequent content error in both datasets concerns omission of information. For the E2E dataset, the *family friendly* attribute is most frequently dropped by both model types, indicating that the verbalization of this boolean attribute is more difficult to learn than other attributes, whose values mostly appear verbatim in the text. Information modification of the word-based model is mainly due to confusing *English* with *Italian* food. Information addition and repetition only occur in the WebNLG dataset. The latter is an especially frequent problem of the character-based model, affecting more than a quarter of all texts.

In comparison, character-based models reproduce the content more faithfully on the E2E dataset while offering the same level of linguistic quality as word-based models, leading to more correct outputs overall. On the WebNLG dataset, the word-based model is more faithful to the inputs, probably because of the effective delexicalization strategy, whereas the character-based model errs less on the linguistic side. Overall, the word-based model yields more correct texts, stressing the importance of delexicalization and data normalization in low resource settings.

6.4 Automatic Evaluation of Output Diversity

While correctness is a necessity in NLG, in many settings it is not sufficient. Often, variation of the generated texts is crucial to avoid repetitive and unnatural outputs. Table 6 shows automatically computed statistics on the diversity of the generated texts of both models and human texts and on the overlap of the (generated) texts with the training set. We measure diversity by the number of unique sentences and words in all development set references and generated texts, as done e.g. by Devlin et al. (2015). Additionally, we report the Shannon text entropy as measure of the amount of variation in the texts following (Oraby et al., 2018b). We compute the text entropy E for words (unigrams) and uni-, bi-, and trigrams as follows:

$$E = - \sum_{w \in V} \frac{f(w)}{\text{total}} * \log_2 \frac{f(w)}{\text{total}} \quad (3)$$

where V is the set of all word types or uni-, bi- and trigrams, f denotes frequency and total is the token count or total number of uni-, bi- and trigrams in the texts, respectively.

To measure the extent by which the models generalize beyond plugging in restaurant or other entity names into templates extracted from the training data, we compute the results on the delexicalized outputs of the word-based models and delexicalize the character-based models’ outputs. For the human scores, we generate n artificial prediction files, treating each n -th reference (42 for E2E, 8 for WebNLG) as reference, apply delexicalization, and average the scores for the n files.

On both datasets, our systems produce significantly less varied outputs and reproduce more

Template 1: NAME is a [FAMILY-FRIENDLY] EATTYPE which serves [FOOD] food [in the PRICE RANGE price range]. [It has a RATING rating] [and is located in the AREA area[, near NEAR]]. [It is not FAMILY-FRIENDLY.]

Example: NAME is a family-friendly coffee shop which serves Chinese food in the low price range. It has a high customer rating and is located in the city centre area, near NEAR.

Template 2: The [FAMILY-FRIENDLY] EATTYPE NAME serves [FOOD] food [in the PRICE RANGE price range]. [It is located in the AREA area[, near NEAR].] [It has a RATING rating.] [It is not FAMILY-FRIENDLY.]

Example: The family-friendly coffee shop NAME serves Chinese food in the low price range. It is located in the city centre area, near NEAR. It has a high customer rating.

Learned combinations of Template 1 and 2:

- NAME is a restaurant which serves English food in the moderate price range. It is located in the city centre area, near NEAR. It has a customer rating of 1 out of 5. It is not family friendly.
- The family-friendly pub NAME serves Indian food in the low price range. It has a customer rating of 5 out of 5 and is located in the riverside area, near NEAR.

Figure 2: Templates used for synthetic training data generation, parts in brackets are realized only if the input contains the corresponding attribute. Learned combinations are two template combinations produced by a model trained on data generated from both templates.

texts and sentences from the training data than the human texts. Interestingly, however, the character-based models generate significantly more unique sentences and copy significantly less from the training data than the word-based models, which copy about 40% of their generated sentences from the training data.

7 Generalizing from Templates

In search for empirical evidence that neural models are able to surpass the structures they were trained on, we train Seq2Seq models with synthetic training data created by templates. This enables us to control the variation in the training data and identify novel generations of the model (if any). We investigate two questions: (1) Do the neural NLG models indeed accurately learn the templates from the training data? (2) Do they learn to combine the training templates to produce more varied outputs than seen during training?

We generate synthetic training data based on two templates. Template 1 corresponds to UKPTUDA’s submission to the E2E challenge⁷, where the order of describing the input information is fixed. Specifically, the restaurant’s customer rating is always mentioned before its location. For Template 2, we change the the beginning of the template and switch the order of mentioning the

rating and location of the restaurant as shown in Figure 2. Potential combinations of the two templates are to combine the beginning of Template 1 with the ordering of rating and area of Template 2 or vice versa. We generate a single reference text for all 2261 training inputs of the E2E dataset where the NAME and EATTYPE attribute are present as these are the two obligatory attributes for the templates. We train word-based models on training data generated with Template 1, Template 2 and the concatenation of the training data from Template 1 and 2. To keep the amount of training data equal in all experiments, we once repeat the training corpus generated only with Template 1 or Template 2. The hyperparameters for the three models can be found in the appendix.

	c@1	c@2	c@5	c@30
template 1	0.8	0.8	0.9	1.7
template 2	1.0	1.2	1.3	1.9
template 1+2	0.9	1.6	2.2	3.3
+ reranker	0.9	1.9	2.7	3.3

Table 7: Manual evaluation of generated texts for 10 random test instances of a word-based model trained with synthetic training data from two templates. c@n: avg. number of correct texts (with respect to content and language) among the top n hypotheses.

⁷<https://github.com/UKPLab/e2e-nlg-challenge-2017/blob/master/components/template-baseline.py>

Table 7 shows our manual evaluation of the top 30 hypotheses for 10 random E2E test inputs generated by models trained with data synthesized from the two templates. As is evident from the first two rows, all models learned to generalize from the training data to produce correct texts for novel inputs consisting of unseen combinations of input attributes. It was verified in the manual evaluation that 100% of the texts generated by models trained on a single template adhered to this template. Yet, the picture is a bit different for the model trained on data generated by both templates. While the top two hypotheses are equally distributed between adhering to Template 1 and Template 2, more than 5% among the lower-ranked hypotheses constitute a template combination such as the example shown in the bottom part of Figure 2. For 60% of the examined inputs, there was at least one such hypothesis resulting from template combination, of which two thirds were actually correct verbalizations of the input.

Since we found that the models frequently ranked correct hypotheses below hypotheses with content errors, we implemented a simple rule-based reranker based on verbatim matches of attribute values. The reranker assigns an error point to each omission and addition of an attribute value. As can be seen in the final row of Table 7, this simple reranker successfully places correct hypotheses higher up in the ranking, improving the practical usability of the generation model by now offering almost three correct variants for each input among the top five hypotheses on average.

8 Conclusion

We compared word-based and character-based Seq2Seq models for data-to-text NLG on two datasets and analyzed their output diversity. Our main findings are as follows: Overall, Seq2Seq models can learn to verbalize structured inputs in a decent way; their success depends on the extent of the domain and available (clean) training data.

Second, in a comparison with texts produced by humans, we saw that neural NLG models can even surpass human performance in terms of automatic evaluation measures. On the one hand, this unveils the ability of the models to extract general patterns from the training data that approximate many reference texts, but on the other hand also once more stresses the limited utility of such measures to evaluate NLG systems.

Third, in light of the multi-faceted analysis we performed, it is difficult to draw a general conclusion on whether word- or character-based processing is more useful for data-to-text generation. Both models yielded comparable results with respect to automatic evaluation measures. In the manual error analysis, the character-based model performed better on the E2E dataset, whereas the word-based model generated more correct outputs on the WebNLG dataset. Character-based models were found to have a significantly higher output diversity.

Finally, in a controlled experiment with word-based Seq2Seq models trained on data synthesized from templates, we showed the capability of such models to perfectly reproduce the templates they were trained on. More importantly, models trained on two templates could generalize beyond their training data and come up with novel texts. In future work, we would like to extend this line of research and train more model variants on a higher number of templates.

References

- Shubham Agarwal and Marc Dymetman. 2017. [A surprisingly effective out-of-the-box char2char model on the e2e nlg challenge dataset](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 158–163, Saarbrücken, Germany.
- Gabor Angeli, Percy Liang, and Dan Klein. 2010. [A simple domain-independent probabilistic approach to generation](#). In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 502–512, Stroudsburg, PA, USA.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, abs/1409.0473.
- Rachel Bawden. 2017. [Machine translation, it’s a question of style, innit? the case of english tag questions](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2507–2512, Copenhagen, Denmark.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O’Reilly Media, Inc.
- Adam Cheyer and Didier Guzzoni. 2006. Method and apparatus for building an intelligent automated assistant. Patent US 11/518,292 (Patent pending).
- Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning](#)

- phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*, pages 1693–1703, Berlin, Germany.
- Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language models for image captioning: The quirks and what works. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Volume 2: Short Papers*, pages 100–105, Beijing, China.
- Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 2: Short Papers*, pages 41–51, Berlin, Germany.
- Jessica Fidler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Volume 1: Long Papers*, pages 179–188, Vancouver, Canada.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG Challenge: Generating Text from RDF Data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain.
- 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 (JAIR)*, 61:65–170.
- Raghav Goyal, Marc Dymetman, and Éric Gaussier. 2016. Natural language generation through character-based rnns with finite-state prior knowledge. In *Proceedings of the 26th International Conference on Computational Linguistics, COLING 2016, Technical Papers*, pages 1083–1092, Osaka, Japan.
- Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*, pages 1631–1640, Berlin, Germany.
- Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017*, pages 252–256, Santiago de Compostela, Spain.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8).
- Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 329–339, Austin, TX, USA.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, San Diego, CA, USA.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810.
- Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *TACL*, 5:365–378.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*, pages 994–1003, Berlin, Germany.
- Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of summaries. In *Proceedings of the ACL workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.
- Grace Lin and Marilyn Walker. 2011. All the worlds a stage: Learning character models from film. In *Proceedings of the Seventh AIIDE Conference*, pages 46–52, Palo Alto, CA, USA.
- Wang Ling, Isabel Trancoso, Chris Dyer, and Alan Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.

- Zachary Chase Lipton, Sharad Vikram, and Julian McAuley. 2015. [Capturing meaning in product reviews with character-level generative text models](#). *CoRR*, abs/1511.03683.
- Thang Luong, Hieu Pham, and Christopher Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1412–1421, Lisbon, Portugal.
- François Mairesse and Steve J. Young. 2014. [Stochastic language generation in dialogue using factored language models](#). *Computational Linguistics*, 40(4):763–799.
- Francois Mairesse and Marilyn Walker. 2008. Trainable Generation of Big-Five Personality Styles through Data-driven Parameter Estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 165–173, Columbus, OH, USA.
- Hongyuan Mei, Mohit Bansal, and Matthew Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*, pages 720–730, San Diego, CA, USA.
- Danilo Mirkovic, Lawrence Cavedon, Matthew Purver, Florin Ratiu, Tobias Scheideck, Fuliang Weng, Qi Zhang, and Kui Xu. 2006. [Dialogue management using scripts and combined confidence scores](#). US Patent App. 11/298,765.
- Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. [A study of style in machine translation: Controlling the formality of machine translation output](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2814–2819, Copenhagen, Denmark.
- 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, EMNLP 2017*, pages 2231–2242, Copenhagen, Denmark.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. [The e2e dataset: New challenges for end-to-end generation](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany.
- Alice Oh and Alexander Rudnicky. 2000. [Stochastic language generation for spoken dialogue systems](#). In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3, ANLP/NAACL-ConvSyst '00*, pages 27–32, Stroudsburg, PA, USA.
- Shereen Oraby, Lena Reed, Sharath T. S., Shubhangi Tandon, and Marilyn A. Walker. 2018a. Neural multivoice models for expressing novel personalities in dialog. In *Interspeech*, pages 3057–3061, Hyderabad, India. ISCA.
- Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T. S., Stephanie Lukin, and Marilyn Walker. 2018b. [Controlling personality-based stylistic variation with neural natural language generators](#). In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 180–190, Melbourne, Australia.
- 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, ACL '02*, pages 311–318, Stroudsburg, PA, USA.
- Nils Reimers and Iryna Gurevych. 2017. [Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark.
- Ehud Reiter and Anja Belz. 2009. [An investigation into the validity of some metrics for automatically evaluating natural language generation systems](#). *Computational Linguistics*, 35(4):529–558.
- Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.
- Donia Scott and Johanna Moore. 2006. An NLG evaluation competition? Eight Reasons to Be Cautious. In *Proceedings of the Fourth International Natural Language Generation Conference, INLG 2006, Special Session on Sharing Data and Comparative Evaluations*, Sydney, Australia.
- Abigail See, Christopher Manning, and Peter Liu. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Association for Computational Linguistics*.
- 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing*, pages 341–351, Berlin, Heidelberg. Springer.
- Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 1017–1024, Bellevue, WA, USA.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14*, Cambridge, Massachusetts, USA. MIT Press.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of the International Conference on Machine Learning, Deep Learning Workshop*, Lille, France.

Marilyn A. Walker, Ricky Grant, Jennifer Sawyer, Grace I. Lin, Noah Wardrip-Fruin, and Michael Buell. 2011. Perceived or Not Perceived: Film Character Models for Expressive NLG. In *ICIDS*, volume 7069 of *Lecture Notes in Computer Science*. Springer.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015a. [Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking](#). *CoRR*, abs/1508.01755.

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

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015b. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1711–1721, Lisbon, Portugal.

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, EMNLP 2017*, pages 2253–2263, Copenhagen, Denmark.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google's neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

Ming Zhou, Mirella Lapata, Furu Wei, Li Dong, Shao-han Huang, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 1: Long Papers*, pages 623–632, Valencia, Spain.

A Hyperparameters for Models Trained on Synthetic Training Data

For the model trained on template-generated data, we tune the hyperparameters to achieve 100% accuracy for their best hypotheses on template-generated references on the development set. All models have a single-layer LSTM with 64 hidden units in the encoder and decoder. We half the learning rate starting from the eighth training epoch or if the perplexity of the validation set does not improve. The gradient norm is capped at two. The decoder uses the general attention mechanism. For decoding, we set the beam size to 30. Table 8 shows hyperparameters which differ for the models.

hyperparameter	T 1	T 2	T 1+2
encoder	unidirectional		bidir.
embedding size	28	28	30
optimizer	Adam	SGD	SGD
init. learning rate	0.001	1.0	1.0
batch size	4	4	16
dropout	0.4	0.5	0.3
epochs	25	13	15

Table 8: Hyperparameters for the models trained on synthetic training data generated from Template 1 (T 1), Template 2 (T 2) and both (T 1+2).