E2E NLG Challenge: Neural Models vs. Templates

Yevgeniy Puzikov and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
Research Training Group AIPHES
www.ukp.tu-darmstadt.de

Abstract

E2E NLG Challenge is a shared task on generating restaurant descriptions from sets of key-value pairs. This paper describes the results of our participation in the challenge. We develop a simple, yet effective neural encoder-decoder model[1] which produces fluent restaurant descriptions and outperforms a strong baseline. We further analyze the data provided by the organizers and conclude that the task can also be approached with a template-based model developed in just a few hours.

MR:

name[The Eagle]	*eatType[coffee shop]*
food[French]	*priceRange[moderate]*
customerRating[3/5]	*area[riverside]*
kidsFriendly[yes]	*near[Burger King]*

Human Natural Language Reference:

The three star coffee shop, The Eagle, gives families a mid-priced dining experience featuring a variety of wines and cheeses. Find The Eagle near Burger King.

Figure 1: E2E NLG Challenge data specification.

1 Introduction

Natural Language Generation (NLG) is the task of generating natural language utterances from structured data representations. The latter can be a syntactic tree (Mani et al., 1999; Barzilay and McKeown, 2005), a set of abstract logic forms (Phillips, 1993), key-value attribute pairs (Chisholm et al., 2017; Dong et al., 2017), vector space embeddings (Dinu and Baroni, 2014), multivariate time series (Sripada et al., 2003), sensor measurements (Yu et al., 2007).

The E2E NLG Challenge[2] is a shared task which focuses on end-to-end data-driven NLG methods. These approaches attract a lot of attention, because they perform joint learning of textual structure and surface realization patterns from non-aligned data, which allows for a significant reduction of the amount of human annotation effort needed for NLG corpus creation (Wen et al., 2015; Mei et al., 2016; Dušek and Jurcicek, 2016; Lampouras and Vlachos, 2016).

Our contribution can be summarized as follows:

- we show how exploiting data properties allows us to design more accurate neural architectures
- we develop a simple template-based system which achieves performance comparable to neural approaches

This section further describes the task, data set and baseline provided by the organizers. Section 2 introduces the two systems we developed. Section 3 describes metric-based evaluation results, followed by error analysis of the systems' predictions. In order to confirm our findings, we take a closer look at the data set and present our observations in Section 4. Finally, we conclude with a discussion section which summarizes the obtained results (Section 6).

1.1 Task Definition

The organizers of the shared task provided a crowd-sourced data set of 50k instances in the restaurant domain (Novikova et al., 2017b). Each training instance consists of a dialogue act-based meaning representation (MR) and up to 16 references in natural language (see Figure 1):

[1] https://github.com/UKPLab/e2e-nlg-challenge-2017

[2] http://www.macs.hw.ac.uk/InteractionLab/E2E

The data was collected using pictorial representations as stimuli, with the intention of creating more natural, informative and diverse human references compared to the ones one might generate from textual inputs.

The task is to generate an utterance from a given MR, which is both similar to human-generated reference texts and highly rated by humans. Similarity is assessed using standard evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015). However, the final assessment is done via human ratings obtained using a mixture of crowdsourcing and expert annotations.

E2E NLG Challenge data set is split into training, validation and testing sets in a 76.5 – 8.5 – 15 ratio, ensuring that MRs in different sets are distinct. Development set inputs were paired with multiple references to facilitate more reliable model selection procedure; test data contained only input MRs. The objective was to develop and train an NLG system and submit restaurant description predictions for each of the test instances.

To facilitate a better assessment of the proposed approaches, the organizing team used TGen (Dušek and Jurcicek, 2016), one of the recent E2E data-driven systems, as a baseline. It is a sequence-to-sequence (seq2seq) neural system with attention (Bahdanau et al., 2014). In addition to the standard seq2seq module, TGen uses beam search for decoding, incorporates a reranker over the top k outputs, penalizing the candidates that do not verbalize all attributes from the input MR. TGen also includes a delexicalization module which deals with sparsely occurring MR attributes (*name*, *near*) by mapping such values to placeholder tokens when preprocessing the input data, and substituting the placeholders with actual values as a post-processing step.

2 Our Approach

This section describes two different approaches we developed for the shared task.

The first one (Model-D, for "data-driven") is an encoder-decoder neural system which is similar to TGen, but uses a more efficient encoder module. The primary limitation of the baseline's architecture that we target is the sequential nature of the encoder. Given a set of MR key-value pairs, TGen linearizes it into a sequence of tokens by concatenating keys and values. The resultant sequence is further fed to a recurrent neural network (RNN). RNNs have an advantage of being able to process variable-sized inputs. However, they also learn dependencies between sequence items, which might not be desired in some cases. We decided to investigate ways of exploiting data properties in order to deal with inputs of different sizes while refraining from imposing any dependencies between the constituting MR attributes, since each input MR is a set of items, not a sequence (Section 2.1).

Section 2.2 introduces the second approach which is a simple template-based model (Model-T, for "template-based") which we developed based on the results of the exploratory data analysis. We view such a system as a necessary candidate for comparison, since the E2E NLG Challenge data was designed to learn models that produce "more natural, varied and less template-like system utterances" (Novikova et al., 2017b).

2.1 Model-D

Model-D was motivated by two important properties of the E2E NLG Challenge data:

- fixed number of unique MR attributes
- low diversity of the lexical instantiations of the MR attribute values

Each input MR contains a fixed number of unique attributes (between three and eight), which allows us to associate a positional id with each attribute and omit the corresponding attribute names (or keys) from the encoding procedure. This shortens the encoded sequence, presumably making the learning procedure easier for the encoder. This also unifies the lengths of input MRs and thus allows us to use simpler and more efficient neural networks which are not sequential and process input sequences in one step (e.g. multilayer perceptron (MLP) networks).

One might argue that using an MLP would be complicated by the fact that neither the number of active (non-null value) input MR keys nor the number of tokens constituting the corresponding values is fixed. For example, an MR key *price* may have a one-token value of "low" or a more lengthy "less than £10". However, realizations of the MR attribute values exhibit low variability: six out of eight keys have less than seven unique values, while the remaining two keys (*name*, *near*) denote named entities and thus are easy to delexicalize.

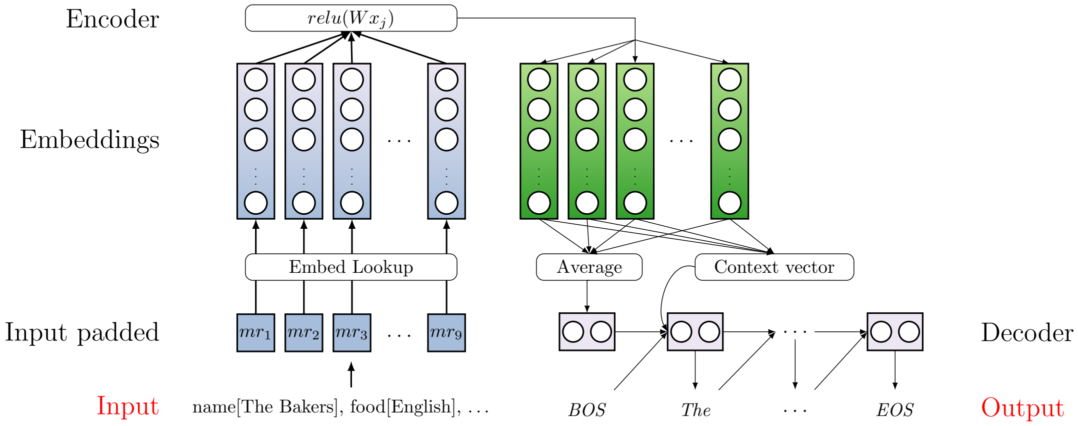

Figure 2: Schematic view of the neural network architecture (Model-D).

This allows us to treat each value as a single token, even if it consists of multiple words (e.g. "more than £30", "Fast food").

Each predicted output is a textual description of a restaurant. Figure 3 shows a histogram of the distribution of the lengths of reference texts in the training data. A reference's length was measured as the number of tokens comprising the reference (including punctuation).[3] We used the value of 50 as a cut-off threshold, filtering out training instances with long restaurant descriptions.

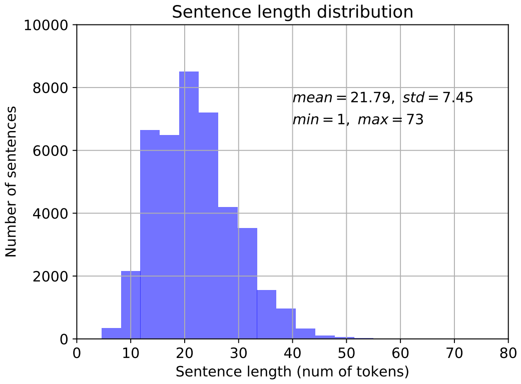

Figure 3: E2E NLG Challenge data analysis: length distribution of restaurant descriptions in the training data.

The overall architecture of our model is shown in Figure 2. The system is an encoder-decoder model (Cho et al., 2014b; Sutskever et al., 2014) consisting of three main modules: an embedding matrix, one dense hidden layer as an encoder and a RNN-based decoder with gated recurrent units (GRU) (Cho et al., 2014a).

posID	*Key*	*Value*
1	*area*	*PAD*
2	*customerRating*	*high*
3	*eatType*	*PAD*
4	*familyFriendly*	*yes*
5	*food*	*PAD*
6	*name*	*Wrestlers*
7	*near*	*PAD*
8	*priceRange*	*PAD*

Table 1: Input representation of the running example using positional ids.

Let us first describe the input specifications of the model. We will use the following MR instance as a running example:

name[Wrestlers] *customerRating[high]*
familyFriendly[yes]

Considering the alphabetic ordering of the MR key names, we can assign positional ids to the keys as shown in Table 1. The remaining five keys are assigned dummy *PAD* values.

Given an instance of a *(MR, text)* pair, we decompose the MR into eight components (mr_j in Figure 2), each corresponding to a value for a unique MR key, and add an end-of-sentence symbol (*EOS*) to denote the end of the encoded sequence. For notation purposes, let us denote the total number of components as $N = 9$ (including *EOS*). Each component is represented as a d-dimensional embedding vector $x_j \in \mathbb{R}^d, j \in \{1, \ldots, N\}$. The embedding matrix which contains all such vectors is denoted as $E \in \mathbb{R}^{d \times |V|}$, where V is the vocabulary of unique tokens observed in the training data. Each embedding vector is further mapped to a dense hidden representation via an affine transformation followed by a ReLu (Nair and Hinton,

[3] As reported by Novikova et al. (2017b), the average number of words per reference is 20.1.

2010) function:

$$h_j = relu(Wx_j) \tag{1}$$

Here $W \in \mathbb{R}^{k \times d}$ is a weight matrix and $h_j \in \mathbb{R}^k$ is a dense representation of the MR component mr_j. We take an average of the encoder outputs and initialize the decoder with the resultant mean vector:

$$s_0 = \frac{1}{N} \sum_{j=1}^{N} h_j \tag{2}$$

Vectors h are further used by the decoder network, which in our case is a unidirectional GRU-based RNN with an attention module (Bahdanau et al., 2014). The decoder generates a sequence of tokens, one token at a time, until it predicts the *EOS* token. At timestep t the decoder defines a probability of generating a token y_t, based on the previously predicted word y_{t-1}, previous hidden state of the GRU unit s_{t-1} and the context vector c_t:

$$p(y_t | y_1, \ldots, y_{t-1}, x) = softmax(g(y_{t-1}, s_t, c_t)) \tag{3}$$

Here g is a transformation function that outputs a vocabulary-sized vector. The hidden state of the decoder is computed as follows:

$$s_t = gru(y_{t-1}, s_{t-1}, c_t) \tag{4}$$

The context vector c_t is a weighted sum of the input representations computed by the encoder:

$$c_t = \sum_{j=1}^{N} \alpha_{tj} h_j \tag{5}$$

Weights α_{tj} are computed as follows:

$$\alpha_{tj} = \frac{\exp{(e_{tj})}}{\sum_{l=1}^{N} \exp{(e_{tl})}}; \tag{6}$$

$$e_{tj} = v_a^T tanh(W_a s_{t-1} + U_a h_j) \tag{7}$$

Here $v_a \in \mathbb{R}^m$, $W_a \in \mathbb{R}^{m \times k}$, $U_a \in \mathbb{R}^{m \times k}$ are weight matrices storing the parameters of the attention module. Our model employs the greedy search decoding strategy and does not use any reranker module.

We train the system for 30 epochs, saving the predictions and performance scores at each epoch. After the training procedure is finished, we choose the model with the highest score as measured on the development set. Since we have five evaluation metrics (provided by the organizers, see Table 2) and it is not clear which one better reflects text quality, we decided to use a simple average of the metrics as a model selection criterion. Before computing the average, the scores for each metric are normalized according to the following formula:

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \tag{8}$$

We did not perform extensive hyper-parameter optimization, all values[4] were set according to what we deemed reasonable on the basis of common practice in the community. We elaborate on the reasons for this decision in the evaluation Section 3.

2.2 Model-T

Taking into consideration low lexical variation of the MR attribute values, one might be interested in whether it is possible to design a deterministic NLG system to tackle the task. We examined the ways MR attribute keys and values are verbalized in the training data and discovered that the majority of textual descriptions follow a similar ordering of MR attribute verbalizations:

```
[name] is a [familyFriendly] [eatType]
which serves [food] food in the [price]
price range. It has a [customerRating]
customer rating. It is located in the
[area] area, near [near].
```

Here `[X]` denotes the value of the MR key `X`. This pattern became a central template of Model-T. Not all MR attribute verbalizations fit into this schema. For example, a key-value pair *customer-Rating[3 out of 5]* would be verbalized as "…has a 3 out of 5 customer rating", which is not the best phrasing one can come up with. A better way to describe it is "…has a customer rating of 3 out of 5". We incorporate such variations into Model-T with a set of simple rules which modify the general template depending on a specific value of an MR attribute.

As mentioned in Section 2.1, each instance's input can have up to eight MR attributes. In order to account for this fact, we decomposed the general template into smaller components, each corresponding to a specific MR attribute mentioned in the input. We further developed a set of rules which activate each component depending on whether an MR attribute is part of the input. For example, if

[4] Can be found in the configuration files in the GitHub repository.

Metric	TGen	Model-D	Model-T
BLEU	0.6925	**0.7128** ± 0.013	0.6051
NIST	8.4781	**8.5020** ± 0.092	7.5257
CIDEr	2.3987	**2.4432** ± 0.088	1.6997
ROUGE-L	0.7257	**0.7378** ± 0.015	0.6890
METEOR	0.4703	**0.4770** ± 0.012	0.4678

Table 2: Evaluation results according to automatic metrics (development set).

price is not in the set of input MR attributes, then the general template becomes:

```
[name] is a [familyFriendly] [eatType]
which serves [food] food. It has a
[customerRating] customer rating.
It is located in the [area] area,
near [near].
```

Finally, we also add a simple post-processing step to handle specific punctuation and article choices.

When designing the templates, we observed that the provided data contains artifacts which cannot be attributed to the occasional noise or annotation guideline decisions made by the organizers. For example, 33 out of 5996 (or 0.55%) of training instances with the input attribute *food[Japanese]* had descriptions of Chinese restaurants as reference outputs. This probably was caused by incorrect interpretation of the input MR – as we mentioned in Section 1.1, the data originally was pictorial, so some crowd workers might have chosen a wrong label for denoting the Japanese cuisine.

Another example is unmotivated specification of price ranges (*priceRange[cheap]* → "the cheap price of £10.50") or customer ratings (*customerRating[high]* → "a 9 on a scale of 1-10"). These cases are most likely the result of some of the annotators' attempt to be creative.

3 Evaluation

3.1 Metric Evaluation

Following the setup of the competition, we analyze the performance of the proposed approaches on the development set using five automatic metrics implemented in the provided scripts[5] (Table 2).

In our comparison, we rely on the performance of TGen reported by the organizers of the shared task (Novikova et al., 2017b). Since we were provided with only one TGen prediction file and a single performance score, comparing score distributions is not possible and statistical significance tests are not meaningful due to the non-deterministic nature of the approaches based on neural networks and randomized training procedures (Reimers and Gurevych, 2017). In order to facilitate a fair comparison with other competing systems, we report the mean development score of Model-D (averaged across twenty runs with different random seeds) and performance variance for each automatic metric. The corresponding twenty predictions are available in our code repository. Model-T is a deterministic system, so it is sufficient to report the results of a single run.

The results show that Model-D outperforms TGen as measured by all five metrics, albeit the performance variance is quite large. Model-T clearly scores below both TGen and Model-D. This is expected, since Model-T is not data-driven, and hence the texts it generates might be different from the reference outputs.

However, this does not yet mean that Model-D is better – metric-based evaluation is just a proxy estimator of the quality of text candidates. Previous studies have shown that widely used automatic metrics (including the ones used in our competition) lack strong correlation with human judgments (Scott and Moore, 2007; Reiter and Belz, 2009; Novikova et al., 2017a). The majority of these metrics come from the fields of machine translation and automatic document summarization and assess content overlap between a reference text and the generated output. They do not measure fluency or discourse structure of the candidate output. Moreover, in cases when a model ignores parts of the input when generating text or hallucinates contents not given in the input, these metrics rely entirely on the quality of the references in their assessment of the candidates. To make more solid conclusions, we performed manual error analysis of the predictions made by the compared systems.

3.2 Error Analysis

We randomly sampled 100 input instances from the development set and retrieved the corresponding outputs from the official baseline prediction file provided by the organizers. After training Model-D we had twenty serialized model instances and prediction files. We randomly picked one instance and retrieved predictions for the sampled 100 inputs. Finally, we also extracted the corresponding

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

Error type	**TGen**	**Model-D**	**Model-T**
dropped contents	9	**49**	0
punctuation errors	1	**12**	0
modified contents	4	4	0
bad grammar	4	1	0

Table 3: Common errors made by the compared models (100 randomly sampled development instances).

predictions of our template-based model.

We focused on generic errors, which make sense to look out for in many NLG scenarios. Table 3 shows the error types and number of mistakes found in each of the prediction files. The error types should be self-explanatory (sample predictions are given in Appendix A.2).

As far as the (subjective) manual analysis goes, Model-T outputs descriptions with the best linguistic quality. Table 3 shows that the predictions of the template-based system contain no errors – this is because we incorporated our notion of grammaticality into the templates' definition, which allowed Model-T to avoid the errors found in predictions of the other two approaches.

Note that although Model-T is able to produce fluent and grammatical restaurant descriptions, it inevitably suffers from low output variety. Its advantage lies in the fact that we can easily adjust it to generate more user-specific texts.

The majority of errors made by Model-D are either wrong verbalizations of the input MR values or punctuation mistakes. The latter ones are limited to the cases of missing a comma between clauses or not finishing a sentence with a full stop. An easy solution to this problem is adding a post-processing step which fixes punctuation mistakes before outputting the text. TGen has an advantage here, since it uses a set of rules which correct punctuation errors, while Model-D is purely data-driven (and as we show in the following section, punctuation errors are common in the provided data).

A more interesting challenge is posed by those cases where our model drops or modifies some MR attribute values. According to the organizers, 40% of the data by design contain either additional or omitted information on the output side (Novikova et al., 2017b): crowd workers were allowed to not lexicalize attribute values which they deemed unimportant. Taking this into consideration, we might conclude that our model outperforms TGen, since Model-D generates texts which are more similar to the ones encountered in the training data. Unfortunately, the exact definition of importance used for the annotation is unknown to us, which is why we could not assess the validity of Model-D's behaviour in specific cases.

To corroborate this hypothesis, we have decided to examine the training data and find out if the discrepancies of Model-D were learned from the data.

4 Training Data Analysis

The E2E NLG Challenge is based on real, noisy data. In contrast to small, but highly curated data sets (BAGEL (Mairesse et al., 2010), SF Hotels/Restaurants (Wen et al., 2015)), the organizers provided multiple instances to account for this noise. In order to better understand the behaviour of Model-D and determine if it took advantage of having multiple references per training instance, we have randomly sampled 100 training instances and manually checked their linguistic quality. Table 4 shows the most common errors we encountered.

Most mistakes come from ungrammatical constructions, e.g. incorrect phrase attachment decisions ("The price of the food is high and is located ..."), incorrect usage of articles ("located in riverside"), repetitive constructions ("Cotto, an Indian coffee shop located in ..., is an Indian coffee shop ..."). Some restaurant descriptions follow a tweet-style narration pattern which is understandable, but ungrammatical ("The Golden Palace Italian riverside coffee shop price range moderate and customer rating 1 out of 5").

A considerable number of instances have restaurant descriptions which contain information that does not entirely follow from the given input MR. These are cases in which input content elements are modified or dropped, which goes in line with what we observed in the outputs of Model-D.[6]

A few instances (10%) contained descriptions which we marked as questionable. They are grammatical, but are phrased in a way which we would rather avoid due to pragmatic and/or stylistic considerations. For example, restaurants which have *familyFriendly[no]* as part of the input MR are often described by crowd workers as "adults-only" establishments, which has an undesirable connota-

[6] The crowd workers were allowed to not verbalize certain fields. We suspect that this freedom could have left an opportunity for potential abuse by some annotators, which is why we view such cases as potentially harmful and include them into the table.

Error type	Example	%
bad grammar	"it's French food falls within a high price range"	15
modified contents	*area[riverside]* → "city centre"	12
dropped contents	*priceRange[high]* → ∅	10
questionable lexicalization	"Adult-only Chinese restaurant, The Waterman, offers top-rated food in the city centre"	9
punctuation errors	"X is a coffee shop and also a Japanese restaurant great for family and close to Crowne Plaza Hotel"	6

Table 4: Data annotation discrepancies (100 randomly sampled training instances).

tion. Finally, it is necessary to mention that some crowd workers followed inconsistent spelling and punctuation rules. The most prevalent cases of the former are those of hyphenating compound modifiers ("family friendly restaurant", "the restaurant is family friendly"), capitalizing MR attributes ("Riverside", "Fast food") and various typos ("neat" instead of "near", "rage" instead of "range"). Punctuation errors were mainly restricted to missing a full stop at the end of a restaurant description or failing to delimit sentence clauses with commas.

The results of manual data analysis show that Model-D indeed generates texts that are similar to the restaurant descriptions in the provided data set. Unfortunately, our data-driven approach is not flexible enough to make use of multiple references; it cannot cancel out the noise present in some training instances. One way of alleviating this problem could be reformulating the loss function to inform the system about the existence of multiple ways of generating a good restaurant description. Given a training instance, Model-D would generate a corresponding candidate text which could be compared to all human references. Each comparison results in computing a certain cost; the gradients could be then computed on the minimal cost among all comparisons.

The approach can be further improved by adding a post-processing step which fixes punctuation and/or occasional spelling errors.

5 Final Evaluation

For the final submission we have chosen Model-T's predictions – despite lower metric scores, they contained most grammatical outputs and kept all input information in the generated text.

The results of the final evaluation on the test data are presented in Table 5. For comparison, we also include the highest reported scores among all the participants (rightmost column). Full comparison of the systems can be found on the shared task website.

	Model-T	Best result
Metric evaluation		
BLEU	0.5657	0.6805
NIST	7.4544	8.7777
METEOR	0.4529	0.4571
ROUGE-L	0.6614	0.7084
CIDEr	1.8206	2.3371
Human evaluation		
Quality	0.228/(2.0, 4.0)/2	0.300/(1.0, 1.0)/1
Naturalness	0.077/(5.0, 10.0)/2	0.211/(1.0, 1.0)/1

Table 5: Final evaluation results on the test set. Human evaluation results have the following format: *score/(range)/cluster*.

In the human evaluation experiments *Quality* was defined as "an overall quality of the utterance, in terms of its grammatical correctness, fluency, adequacy and other important factors". *Naturalness* was defined as "the extent to which the utterance could have been produced by a native speaker". The final evaluation results were produced by the TrueSkill algorithm (Sakaguchi et al., 2014), which performs pairwise system comparisons and clusters them into groups. The numerical scores are not directly interpretable, but the relative ranking of a system in terms of its range and cluster is important: systems within one cluster are considered tied. Model-T was assigned to the second best cluster both in terms of quality and naturalness, despite the much lower metric scores.

6 Discussion

There are several important conclusions and observations we would like to share:

6.1 Evaluation Criteria

Metric evaluation results gave us an impression that Model-D is much better than both TGen and Model-T. However, manual inspection revealed that all systems have their strong and weak sides. The template-based system produces the most grammatical results, but suffers from low output variety. Model-D and TGen generate more variable outputs which, nevertheless, occasionally contain grammatical errors. Both TGen and Model-T try to verbalize every input MR value, but the organizers expected the systems to perform some content filtering. So, which model should be chosen to solve the task at hand?

Different NLG tasks have different requirements. For example, low diversity of the generated candidates could be an issue for a chat-bot application, but a fact retrieval system is oblivious to that. We viewed the task of generating restaurant descriptions as a purely information-seeking real-world scenario. This is probably not exactly what the task was designed for, since the latter encourages more diverse and less template-like system utterances. Nevertheless, we decided to follow the generic NLG requirements (grammaticality) and submit Model-T's predictions which produced grammatical outputs while keeping all input information in the generated text.

6.2 Development Costs vs. Quality Trade-off

This point is related to the previous one. We acknowledge the importance of developing data-driven models for solving complex problems. However, considering the trade-off between system building cost and output quality, we decided to develop a simple template-based model and compare it to neural architectures.

We spent roughly three hours designing and debugging the template model. It gives consistent, reasonable and fluent output, which can be easily tailored to a particular user by adjusting the templates' contents. On the other hand, Model-D took us approximately a month to develop and several days to optimize. Yet, both models generated texts of comparable linguistic quality.

The E2E NLG Challenge focuses on end-to-end data-driven NLG methods, which is why systems like Model-T might not exactly fit into the task setup. Nevertheless, we hope that our observations and findings facilitate a better understanding of the advantages and disadvantages of various NLG approaches.

6.3 Crowd-sourcing and Business Sensitivity

Consider the following hypothetical prediction candidates:

- "The Bakers is a restaurant serving English food."
- "The Bakers is a restaurant at the riverside, near The Wrestlers."

The two predictions are complementary in terms of their contents and are equal in terms of linguistic quality (fluency). Which one is better? The first sentence mentions the type of the food served, but omits the restaurant's location; the second candidate does the opposite. However, both outputs would probably be rated equally by human assessors in the current task setup, even though we do not know which contents could be dropped and which should be kept intact.

Here the problem we are concerned about is not the question whether to separate content selection from surface realization or not. The issue is that the optimal output of an NLG system is context-sensitive. The task at hand is generating restaurant descriptions, which implies a certain degree of domain-specific "business sensitivity" which not all crowd workers are concerned about. A user looking for family-friendly restaurants might be interested in family-friendly restaurants only. If a restaurant recommendation application decides to omit this information, the user will be very unsatisfied, which has direct implications in business (Levin et al., 2017).

7 Conclusion

In this paper we have presented the results of our participation in the E2E NLG Challenge. We have developed two conceptually different approaches and analyzed their performance, both in quantity and in quality. Our observations and conclusions shed some light on the limitations of modern NLG approaches and possible ways of overcoming them.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under grant No. GU 798/17-1 and the DFG-funded research training group "Adaptive Preparation of Information form Heterogeneous Sources" (AIPHES, GRK 1994/1). We thank our colleagues Michael Bugert, Tristan Miller, Maxime Peyrard, Nils Reimers and Markus Zopf who provided insightful comments and suggestions that greatly assisted our research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics* 31(3):297–328. https://doi.org/10.1162/089120105774321091.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to Generate One-sentence Biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 633–642. http://www.aclweb.org/anthology/E/E17/E17-1060.pdf.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar, pages 103–111. http://www.aclweb.org/anthology/W14-4012.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. 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)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D/D14/D14-1179.pdf.

Georgiana Dinu and Marco Baroni. 2014. How to Make Words with Vectors: Phrase Generation in Distributional Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 624–633. http://www.aclweb.org/anthology/P14-1059.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pages 138–145. http://www.mt-archive.info/HLT-2002-Doddington.pdf.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, 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: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 623–632. http://www.aclweb.org/anthology/E17-1059.

Ondřej Dušek and Filip Jurcicek. 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 (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 45–51. http://www.aclweb.org/anthology/P/P16/P16-2008.pdf.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation Learning for Language Generation from Unaligned Data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 1101–1112. http://www.aclweb.org/anthology/C/C16/C16-1105.pdf.

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*. Association for Computational Linguistics, Prague, Czech Republic, pages 228–231. http://www.aclweb.org/anthology/W/W07/W07-0734.pdf.

Pavel Levin, Nishikant Dhanuka, Talaat Khalil, Fedor Kovalev, and Maxim Khalilov. 2017. Toward a Full-scale Neural Machine Translation in Production: the Booking.com Use Case. *CoRR* abs/1709.05820. http://arxiv.org/abs/1709.05820.

Chin-Yew Lin. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, Barcelona, Spain, pages 74–81. http://www.aclweb.org/anthology/W/W04/W04-1013.pdf.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based Statistical Language Generation Using Graphical Models and Active Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, pages 1552–1561. http://www.aclweb.org/anthology/P/P10/P10-1157.pdf.

Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving Summaries by Revising Them. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, College Park, Maryland, pages 558–565. http://www.aclweb.org/anthology/P/P99/P99-1072.pdf.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to Talk About and How? Selective Generation Using LSTMs with Coarse-to-fine Alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 720–730. http://www.aclweb.org/anthology/N/N16/N16-1086.pdf.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*. Omnipress, USA, pages 807–814.

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*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2231–2242. http://aclweb.org/anthology/D17-1238.

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*. Association for Computational Linguistics, Saarbrücken, Germany, pages 201–206. http://www.aclweb.org/anthology/W/W17/W17-5525.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318. http://www.aclweb.org/anthology/P02-1040.pdf.

John D. Phillips. 1993. Generation of Text from Logical Formulae. *Machine Translation* 8(4):209–235. http://www.jstor.org/stable/40007970.

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 (EMNLP)*. Association for Computational Linguistics, Copenhagen, Denmark, pages 338–348. http://www.aclweb.org/anthology/D/D17/D17-1035.pdf.

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. https://doi.org/10.1162/coli.2009.35.4.35405.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient Elicitation of Annotations for Human Evaluation of Machine

Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 1–11. http://www.aclweb.org/anthology/W14-3301.

Donia Scott and Johanna Moore. 2007. An NLG Evaluation Competition? Eight Reasons to Be Cautious. In *Proceedings of the NSF Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*. National Science Foundation, pages 22–23. http://www.ling.ohio-state.edu/nlgeval07/papers/Scott-Moore.pdf.

Somayajulu G. Sripada, Ehud Reiter, and Ian Davy. 2003. SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. *Expert Update* 6(3):4–10.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, pages 4566–4575. https://doi.org/10.1109/CVPR.2015.7299087.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1711–1721. http://www.aclweb.org/anthology/D/D15/D15-1199.pdf.

Jin Yu, Ehud Reiter, Jim Hunter, and Chris Mellish. 2007. Choosing the Content of Textual Summaries of Large Time-series Data Sets. *Natural Language Engineering* 13(1):25–49. https://doi.org/10.1017/S1351324905004031.

A Supplemental Material

A.1 Manual Data Analysis Results

As mentioned in Section 4, manual analysis of the training data revealed certain annotation problems. Below we provide sentence IDs of the instances which we considered as containing errors:

- modified contents: 4136, 34141, 32915, 35936, 6152, 2005, 1463, 14529, 14339, 21804, 25779, 11049;
- dropped contents: 4136 (price and food), 17455 (familyFriendly), 38742 (area), 1463 (customerRating), 27664 (food, priceRange), 19882 (priceRange), 25779 (customerRating, area), 40293 (food), 807 (familyFriendly), 15216 (familyFriendly);
- ungrammatical: 39810, 1904, 36233, 33671, 28698;
  - tweet-style descriptions: 19425, 38875, 14873;
  - word attachment problems, bad phrasing: 15276, 30950, 31973, 36468, 12442, 18623, 25773;
- Punctuation errors: 6152, 14993, 38742, 14339, 35967, 25795;
- Questionable realization: customerRating (8806, 24983, 29189), familyFriendly (20220, 10729), priceRange (29189), style (14993, 28362, 32494).

A.2 Example Predictions

The majority of predictions from all three systems are grammatically correct. Here is a random sample:

- **Input:** name[Cocum], eatType[coffee shop], food[English], priceRange[moderate], customerRating[3 out of 5], familyFriendly[no]
- **TGen:** Cocum is a moderately priced coffee shop serving english food. It has a customer rating of 3 out of 5 and is not kid friendly.
- **Model-D:** Cocum is a moderately priced coffee shop serving English food . It is not kid friendly and has a customer rating of 3 out of 5 .
- **Model-T:** Cocum is a coffee shop which serves English food in the moderate price range. It has a customer rating of 3 out of 5. It is not family friendly.

To illustrate common prediction mistakes (regardless of the system), we have hand-picked some samples. Full predictions by all models can be found in the GitHub repository.

A.2.1 Bad Grammar

- **Input:** name[The Wrestlers], eatType[coffee shop], food[Chinese], priceRange[less than 20], area[riverside], familyFriendly[no], near[Raja Indian Cuisine]
- **TGen:** The Wrestlers is a coffee shop providing chinese food in the less than 20 price range. It is located in the riverside. It is near Raja Indian Cuisine. Its no children feel allowed.
- **Model-D:** The Wrestlers is a coffee shop providing Chinese food in the less than 20 price range . It is located in the riverside . It is near Raja Indian Cuisine .
- **Model-T:** The Wrestlers is a coffee shop which serves Chinese food in the price range of less than 20. It is located in the riverside area, near Raja Indian Cuisine. It is not family friendly.

- **Input:** name[Wildwood], eatType[coffee shop], food[English], priceRange[high], customerRating[1 out of 5], near[Ranch]
- **TGen:** Wildwood is a high priced coffee shop that serves english food. It is located near Ranch and has a customer rating of 1 out of 5.
- **Model-D:** Wildwood is a coffee shop near Ranch , serving English food , has a high price range and a customer rating of 1 out of 5 .
- **Model-T:** Wildwood is a coffee shop which serves English food in the high price range. It has a customer rating of 1 out of 5 and is located near Ranch.

A.2.2 Modified and/or Dropped Contents

- **Input:** name[The Golden Palace], eatType[coffee shop], food[English], priceRange[high], customerRating[average], area[riverside]

- **TGen:** The Golden Palace is a high priced coffee shop serving english food in the riverside area. It has an average customer rating.

- **Model-D:** The Golden Palace is a high priced coffee shop located in the city centre .

- **Model-T:** The Golden Palace is a coffee shop which serves English food in the high price range. It has an average customer rating and is located in the riverside area.

A.2.3 Dropped Contents

- **Input:** name[The Eagle], eatType[coffee shop], food[English], priceRange[moderate], customerRating[1 out of 5], area[riverside], familyFriendly[yes], near[Burger King]

- **TGen:** The Eagle is a moderately priced coffee shop that serves english food. It is located in the riverside area near Burger King. It is kid friendly and has a customer rating of 1 out of 5.

- **Model-D:** The Eagle is a moderately priced coffee shop located near Burger King .

- **Model-T:** The Eagle is a family-friendly coffee shop which serves English food in the moderate price range. It has a customer rating of 1 out of 5 and is located in the riverside area, near Burger King.