The E2E NLG Challenge: A Tale of Two Systems

Charese Smiley[1], **Elnaz Davoodi** [2], **Dezhao Song**[1], **Frank Schilder**[1]
[1] Research & Development, Thomson Reuters, 610 Opperman Drive, Eagan, MN, USA
[2] Center for Cognitive Computing, Thomson Reuters, 120 Bremner Blvd, Toronto, ON, M5J 3A8, CA
firstname.lastname@thomsonreuters.com

Abstract

This paper presents the two systems we entered into the 2017 E2E NLG Challenge: TemplGen, a templated-based system and SeqGen, a neural network-based system. Through the automatic evaluation, SeqGen achieved competitive results compared to the template-based approach and to other participating systems as well. In addition to the automatic evaluation, in this paper we present and discuss the human evaluation results of our two systems.

1 Introduction

This paper describes our two primary systems for the 2017 E2E NLG Challenge (Novikova et al., 2017): (1) TemplGen, a template-based system that automatically mined templates and (Smiley et al., 2018) (2) SeqGen, a neural network system based on the encoder-decoder architecture (Davoodi et al., 2018).

This NLG challenge involves taking a meaning representation (MR) as input and generating natural language output from it. Many existing NLG systems are template-based because it is easier to control the correctness and the grammaticality of the generated text. For this competition, we explored two approaches, i.e., template-based and neural network-based, in order to examine whether a sequence-to-sequence model based on neural networks would produce better results than a template-based system.

We first briefly introduce the challenge and its dataset in Section 2. We then present the details of our two systems in Section 3. We demonstrate and discuss the results of our systems in Section 4 and conclude in Section 5.

2 The E2E NLG Challenge

The E2E NLG challenge is concerned with the restaurant domain and the dataset was crowdsourced via CrowdFlower (Novikova et al., 2016). The crowdsourced dataset consists of 50,602 instances derived from 5,751 unique MRs (Novikova et al., 2017), and it is larger than previous end-to-end datasets, such as BAGEL (Mairesse et al., 2010) and SF Hotels/Restaurants (Wen et al., 2015).

While creating this dataset, crowd workers were asked to create a verbalization based on a given MR. They were allowed to omit information that they did not find useful. Each MR could contain three to eight different attributes selected from all available attributes: *name, eat type, food, price range, customer rating, area, family friendly*, and *near*. In 40% of the instances, verbalizations contain either omissions or additional information. The dataset is split in a 76.5/8.5/15 ratio into training, development, and test.

The following sample shows a MR and its corresponding natural language (NL) output:

MR:
name [Alimentum],
area [city centre],
familyFriendly [no]

NL:
There is a place in the city centre, Alimentum, that is not family-friendly.

3 Our Two Participating Systems

Many NLG systems are based on templates because the system developers can relatively easily control the system to ensure both grammatical and semantic correctness. However, due to the lack of variability of the used templates, such systems

Proceedings of The 11th International Natural Language Generation Conference, pages 472–477,
Tilburg, The Netherlands, November 5-8, 2018. ©2018 Association for Computational Linguistics

may produce language that does not sound natural and is often perceived as repetitive.

Another type of NLG system is based on neural networks. Although such systems have shown promising results, training such models/systems requires a large amount of training data. Furthermore, neural network-based systems may produce ungrammatical text often with repetitions in the same sentences. There is also no guarantee that the generated text is actually factually correct.

3.1 TemplGen : a Template-based System

First, we delexicalized the data using string match to automatically replace the attribute values contained in the MR with the attribute name (Oraby et al., 2017). The attribute values were largely consistent, but for additional coverage we added a few fuzzy matching rules to account for variants such as *Crown* vs. *Crowne*. We did this for all attributes except for *family friendly* which has a wide range of potential realizations (e.g., positive: *children are welcome, kid friendly*, or negative: *adult only, not for kids*). Therefore, we use a binary yes/no value for that attribute. For each delexicalized sentence, we check to see whether all attributes in the MR were captured during the delexicalization process. If there is a difference between the number of attributes in the MR and the number that were successfully delexicalized, we discard that instance. In total, we discarded roughly 45% of the training sentences. This is slightly more than the 40% of instances in the data that contained omissions or additions. We then use the delexicalized templates to create a dictionary look-up of the MRs.

With the templates now identified, we identify templates that are composed of multiple sentences and split along sentence boundaries. The individual sentences are then stored as partial templates along with the attributes reverse engineered from the templates. Table 1 shows the original template containing 2 sentences and the derived partial templates containing one sentence each. Through this process we collect templates containing all 8 of the attributes individually as well as combinations from 2-8. By extracting individual templates for each attribute alone, we guarantee that we can cover any combination of attributes by generating up to 8 separate sentences although this would not sound very natural.

In the testing phase, we are supplied with an MR which may consist of an unseen combination of attributes. We treat the attributes of the MR as a set, filling the templates using an algorithm that selects the best fitting template.

All templates in the candidate set are relexicalized with the current MR. From there we filter candidates by performing basic sentiment analysis using the NLTK[1] implementation of the VADER sentiment analysis tool (Gilbert, 2014) and removing sentences whose sentiment is incongruent (e.g., great restaurant described as having low rating). To determine this, we look for sentences with non-neutral scores for both positive and negative polarities but no word indicating a reversal such as *however*. The final output from the candidate set is selected at random.

3.2 The sequence-to-sequence system

Our sequence-to-sequence system consists of three main components: Delexicalization, Seq-to-Seq model, and Relexicalization.

Delexicalization. One of the challenges in NLG is generating both semantically and grammatically accurate texts. In order to train a satisfying seq-to-seq model, it is often required to have a large amount of parallel texts. However, among the attributes of the E2E data, most of the non-categorical attributes are very sparse which makes the learning process difficult. Thus, in order to generate accurate sentences based on the meaning representations, we delexicalized the values of some of the attributes to avoid data sparsity (Mairesse et al., 2010; Wen et al., 2015).

The delexicalization process involves replacing the values of the attributes with placeholders. Among the E2E attributes, we delexicalized the values of the attributes which seem to take a value from an open set of values. These include *name*, *price range*, *customer rating* and *near*. We delexicalized both the meaning representations and their corresponding natural language sentences. Delexicalizing *price range* and *customer rating* is more challenging than the others because both attributes have more value variations in the meaning representations and the natural language texts than the other attributes do. Hence, the learning task is between a MR template and a NL template.

Table 2 shows an example of a delexicalized meaning representation and its corresponding delexicalized natural language sentence. The

[1] http://www.nltk.org

	Attributes	Template
Original	customer rating, name, eatType, food, near, area	With a rating of _CUSTOMER RATING_, _NAME_ _EATTYPE_ serves _FOOD_ food. It is located near _NEAR_ and _AREA_.
Partial 1	customer rating, name, eatType, food	With a rating of _CUSTOMER RATING_, _NAME_ _EATTYPE_ serves _FOOD_ food.
Partial 2	near, area	It is located near _NEAR_ and _AREA_.

Table 1: Partial templates extracted from training data.

delexicalized meaning representations are used as input of our Sequence-to-Sequence model, in which the delexicalized natural language sentences are the model target output.

Seq-to-Seq Model. Neural Machine Translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014) is an end-to-end approach for machine translation. Sequence-to-Sequence models adopt the encoder-decoder architecture, in which an input sequence is encoded by the encoder and the output sequence is generated by the decoder (Jean et al., 2014; Luong et al., 2014; Sennrich et al., 2016).

In this challenge, we considered the task as a translation problem which takes a sequence of tokens (i.e., delexicalized meaning representations) as input, and generates a sequence of tokens (i.e., delexicalized natural language sentences) in the same language. In our current implementation, we used the state-of-the-art neural machine translation model (Britz et al., 2017). Table 3 shows the parameters of SeqGen model.

Relexicalization. As a last step, the placeholders in the automatically generated delexicalized sentences should be replaced by their actual values. Thus, for the training and development set, we kept the values of the attributes as they appeared in the original sentences and relexicalized the placeholders with these values. Since there is no corresponding sentence for meaning representations of the test sets, we used the value of the placeholders as they appeared in the original meaning representation. This may have a negative impact on the quality and naturalness.

4 Results and Discussion

4.1 Results

Evaluation for the E2E was conducted using both automatic metrics and human scoring. These results are given in Table 4 with the automatic scoring described in Section 4.2 and the human evaluation in Section 4.3.

4.2 Automatic Scoring

Table 4 shows the results comparing the baseline system with the results from our systems. Systems were evaluated automatically using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Denkowski and Lavie, 2014), ROUGE_L (Lin, 2004), and CIDEr (Vedantam et al., 2015). The first column contains the results for the BASELINE system – a sequence-to-sequence model with attention (Dušek et al., 2018). The other two columns in the table contain the automatic scores for our system where the results for TemplGen is composed of both the training and development data and SeqGen which is the sequence-to-sequence model with beam search decoder with beam size of 5. We tried a beam search decoder with various beam sizes and observed that the search decoder with beam size of 5 achieved the best results compared to the search decoder with larger beam sizes as well as the decoder with no beam search.

For the automatic metrics, none of our systems outperformed the baseline system. However, the SeqGen system outperforms TemplGen and exhibits similar performance to the baseline model which is also sequence-to-sequence based.

4.3 Human Evaluation

For the human evaluation metric (Dušek et al., 2018), raters were shown the reference sentence along with 5 generations from various competing systems. They were asked to rank the generations for *quality* and *naturalness*. For *quality*, raters were given the MR along with the system reference output. They were asked to rank the output based on grammatical correctness, fluency, adequacy, and so on. *Naturalness* measures whether the utterance could have been written by a native speaker. Raters were not given the MR for the

Original Meaning Representation	Original Natural Language Sentences
name [The Rice Boat], food [Indian], area [city centre], near [Express by Holiday Inn]	The Rice Boat is an Indian restaurant in the city centre near the Express by Holiday Inn
Delexicalized Meaning Representation	**Delexicalized Natural Language Sentences**
name [*name_x*], food [Indian], area [city centre], near [*near_x*]	*name_x* is an Indian restaurant in the city centre near *near_x*

Table 2: An example of the delexicalized MR and its corresponding natural language sentence.

Hyper-parameter	Parameter value
Batch size	16
# of hidden units	256
# of encoder layers	3
# of decoder layers	1
RNN cell	GRU
Optimizer	Adam
Input Dropout	1.0
Output Dropout	0.5

Table 3: The list of hyper-parameters tuned for SeqGen model.

Metric	Baseline	TemplGen	SeqGen
BLEU	0.6593	0.4202	0.6336
NIST	8.6094	6.7686	8.1848
METEOR	0.4483	0.3968	0.4322
ROUGE_L	0.6850	0.5481	0.6828
CIDEr	2.2338	1.4389	2.1425
Quality	1 of 5	3 of 5	4 of 5
Naturalness	1 of 5	5 of 5	3 of 5

Table 4: The results of automatic and human evaluation on the test set.

naturalness evaluation. Thus, this metric does not take into account faithfulness to the MR. The results of the human evaluation are based on the system's inferred TrueSkill score (Sakaguchi et al., 2014) which is computed based on pairwise comparisons between systems. For quality, TemplGen out scored SeqGen ranking 3rd and 4th, respectively, out of 5 clusters of systems. The results are reversed for naturalness, with SeqGen performing better than TemplGen ranking 3rd and 5th, respectively, out of 5 clusters. Systems within each cluster are considered statistically indistinguishable.

4.4 Discussion

For the human evaluations, we scored in the third cluster of groups. For a production NLG system, quality of generations would be the most important metric as users would expect faithfulness to the underlying data along with other standards such as grammaticality. Because a variety of features are encompassed within the metric of *quality* (e.g., fluency and adequacy), it is difficult to determine where our algorithm succeeds and fails. To help determine this, we took a closer look at 25 sentences generated by our SeqGen and TemplGen models.

Although SeqGen is an end-to-end approach, there are some limitations. First, we do not have control of the decoder to make it generate all the attributes that appeared in the meaning representations. Thus, the model may suffer from not generating all the attributes or generating extra attributes. In both cases, the relexicalization component either cannot relexicalize all the placeholders or there are extra placeholders that cannot be relexicalized. The first row in Table 5 shows an example of SeqGen generated sentence in which it does not generate text corresponding to all the attributes in the MR. Another issue that we noticed in the generated sentences by SeqGen model is related to the quality of the natural language sentences in terms of grammatically correctness. This issue may be due to the delexicalization. The scond row in Table 5 shows an example where the sentence is grammatically incorrect. In this example, the issue is caused by *less than £20ly priced*.

As for TemplGen, overall, we find that it did not perform well when evaluated on the automatic metrics. This is not particularly surprising as we did not make any attempts to optimize TemplGen for these metrics.

On the other hand, our analysis of the sentences generated by the TemplGen model shows that every generation contained all of the attributes from the MR while one contained additional information. However, we noticed a few issues with fluency that may have contributed to lowering the

Examples generated by SeqGen model		
#	MR	NL
1	name[The Phoenix], eatType[restaurant], food[French], priceRange[cheap], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]	*The Phoenix is a family friendly restaurant located near Raja Indian Cuisine.*
2	name[The Phoenix], eatType[restaurant], food[Fast food], priceRange[less than £20], area[city centre], familyFriendly[no], near[Raja Indian Cuisine]	*The phoenix is a less than £20ly priced fast food restaurant located in the city centre near Raja Indian Cuisine. It is not child friendly.*
Examples generated by TempGen model		
1	name[The Waterman], eatType[restaurant], food[Indian], priceRange[moderate], area[city centre], familyFriendly[yes], near[Raja Indian Cuisine]	*The Waterman is a child friendly restaurant which serves moderately priced Indian food. It is near Raja Indian Cuisine in the city centre area.*
2	name[The Plough], eatType[pub], food[Chinese], priceRange[high], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine],	*The Plough, a Chinese pub near Raja Indian Cuisine, offers family friendly riverside dining for high.*

Table 5: Sample natural language output of MRs generated by both models.

scores for *quality*. Of the suboptimal generations, some were perhaps less fluent due to being composed of multiple sentences as a result of our strategy of randomly selecting a partition of attributes that satisfies the MR. Prioritizing partitions that encompass more attributes may be a simple solution. Finally, other generations sound disfluent due to issues with word choice such as the selection of the partial template containing just an adjective to insert *high* from the MR where a noun phrase such as *a high price* would have sounded more natural (cf. last row in Table 5).

For future work, we will put some restrictions on the decoder such that it would not generate repetitive tokens (including placeholders) and also push the model to generate all the attributes mentioned in the corresponding meaning representation.We also plan to use the released data set for generating semantically similar sentences for the meaning representations.

5 Conclusions

In this paper, we described our two systems for the 2017 E2E challenge: a rule-based system and a sequence-to-sequence neural network system. Although our rule-based system did not score well by automatic metrics, it was able to deliver sentences which are faithful to their underlying MR. On the other hand, our sequence-to-sequence system was also able to achieve decent performance compared to other participating systems.

References

D. Britz, A. Goldie, T. Luong, and Q. Le. 2017. Massive Exploration of Neural Machine Translation Architectures. *arXiv preprint arXiv:1703.03906*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Elnaz Davoodi, Charese Smiley, Dezhao Song, and Frank Schilder. 2018. The E2E NLG Challenge: Training a Sequence-to-Sequence Approach for Meaning Representation to Natural Language Sentences. In *(in prep. for INLG conference)*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

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*, pages 138–145. Morgan Kaufmann Publishers Inc.

Ondrej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *(in prep.)*.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1700–1709.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

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*, pages 1552–1561. Association for Computational Linguistics.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany. ArXiv:1706.09254.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing nlg data: Pictures elicit better data. *arXiv preprint arXiv:1608.00339*.

Shereen Oraby, Sheideh Homayon, and Marilyn Walker. 2017. Harvesting creative templates for generating stylistically varied restaurant reviews. In *Proceedings of the Workshop on Stylistic Variation*, pages 28–36.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *WMT@ ACL*, pages 1–11.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Charese Smiley, Elnaz Davoodi, Dezhao Song, and Frank Schilder. 2018. The E2E NLG Challenge: End-to-End Generation through Partial Template Mining. In *(in prep.)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.