The E2E NLG Challenge: Training a Sequence-to-Sequence Approach for Meaning Representation to Natural Language Sentences

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

Abstract

This paper describes one of Thomson Reuters' primary submissions to the E2E NLG Challenge-2017 shared task. The challenge is focused on end-to-end data-driven natural language generation to learn sentences from non-aligned data. We used a state-of-the-art sequence-to-sequence method to generate natural language sentences from meaning representations. Our automatically generated sentences were evaluated both intrinsically and extrinsically.

1 Introduction

In this paper, we report on the development and results of one of our meaning representation to natural language sentences (MR-to-NL) systems for the E2E NLG Challenge 2017 shared task[1]. We utilized a neural network architecture that performs a sequence-to-sequence translation from an MR template to a natural language output template.

Traditionally, the task of generating human-readable sentences from meaning representations (MR) has focused on two main aspects of language: (1) syntax, and (2) lexicalization. In order to formally formulate this problem, the *sentence planning* subtask focuses on the sentence structure and the *surface realization* subtask corresponds to choosing proper word forms (Reiter and Dale, 2000). An end-to-end NLG model cannot be achieved if any of these subtasks fail. These two subtasks can either be considered as two independent components of an NLG model (Walker et al., 2001; Rieser et al., 2010; Dethlefs et al., 2013), or they can be combined to jointly form one component of the model (Wong and Mooney, 2007; Konstas and Lapata, 2013).

[1]The other primary system is described in (Smiley et al., 2018).

The growing interest in applying deep learning methods to natural language technologies drew our attention to exploring a potential end-to-end deep learning-based solution for this NLG task. Thus, we avoid doing the semantic alignment between the meaning representations and the corresponding sentences in natural languages (NL). Sequence-to-sequence deep learning models (Sutskever et al., 2014) generate an output sequence directly from an input sequence. Machine translation is an example application where these models have shown to outperform traditional approaches (Britz et al., 2017).

2 The E2E Dataset

The E2E dataset (Novikova et al., 2017) contains 42,061 pairs of <meaning representation, natural language sentence(s)> in the training set, and 4,672 pairs in the development set. In this dataset, there are eight different attributes, including *name*, *eat type*, *price range*, *customer rating*, *near*, *food*, *area*, and *family friendly*. Each meaning representation can contain 3 to 8 of these attributes.

3 Model Architecture

As shown in Figure 1, our system consists of three main components:

- De-lexicalization: both the meaning representation and the corresponding target sentences are de-lexicalized.
- Seq-to-Seq model: a de-lexicalized meaning representation is used to generate de-lexicalized natural language sentence(s).
- Re-lexicalization: the generated de-lexicalized sentences are re-lexicalized.

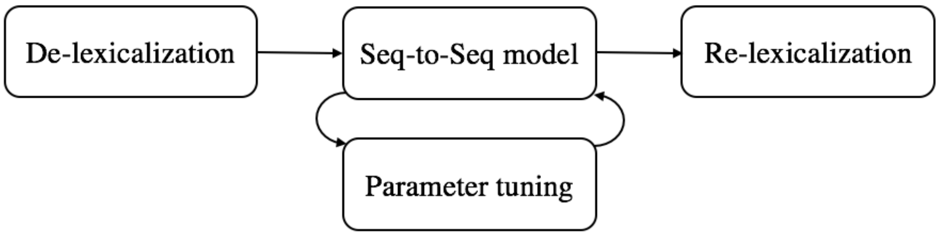

Figure 1: Overview of our MR-to-NL system

3.1 Preprocessing: De-lexicalization of the Meaning Representations and the Natural Language Sentences

One of the challenges in NLG is generating accurate texts which reflect the ground truth (i.e. the fact in a knowledge base of a given domain). Having enough large parallel texts to train a Sequence-to-Sequence model is necessary to generate texts which reflect to the ground truth. 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 de-lexicalized the values of some of the attributes to avoid data sparsity. The de-lexicalization process involves replacing the values of the attributes with placeholders. Among the E2E attributes, we de-lexicalized 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 de-lexicalized both the meaning representations and their corresponding natural language sentences. De-lexicalizing *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. Figure 2 shows an example of a de-lexicalized meaning representation and its corresponding de-lexicalized natural language sentence. The de-lexicalized meaning representations are used as input of our Sequence-to-Sequence model, in which the de-lexicalized natural language sentences are the model target output.

3.2 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 are encoder-decoder models, in which an input sequence (e.g. sequence of tokens in one language) is encoded by the encoder and the output sequence (e.g. sequence of tokens in another language) 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., de-lexicalized meaning representations) as input, and generates a sequence of tokens (de-lexicalized 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).

3.3 Post Processing: Re-lexicalization of the Automatically Generated Sentences

As the last step of our approach, the placeholders in the automatically generated de-lexicalized 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 re-lexicalized 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 Experiments

Based on the model architecture given in section 3, we train two models, each with two variations. We apply the same de-lexicalization and re-lexicalization processes to both models and their variations. The first model (*Model #1*) uses the de-lexicalized meaning representations as the input, and de-lexicalized sentences as target output. The two variations of this model are different in decoding: one variation uses beam search decoder,

Original Meaning Representation	Original Natural Language Sentences
name [The Rice Boat], food [Indian], priceRange [€20-25], customer rating [high], area [city centre], familyFriendly [yes], near [Express by Holiday Inn]	The Rice Boat is an Indian restaurant in the city centre near the Express by Holiday Inn, it is kid friendly highly rated and costs 20-25 euros.

De-lexicalized Meaning Representation	De-lexicalized Natural Language Sentences
name [*name_x*], food [Indian], priceRange [*priceRange_x*], customer rating [*customerRating_x*], area [city centre], familyFriendly [yes], near [*near_x*]	*name_x* is an Indian restaurant in the city centre near *near_x*, it is kid friendly *customerRating_x* rated and costs *priceRange_x*.

Figure 2: An example of the de-lexicalized meaning representation and its corresponding natural language sentence.

	Model #1	Model #2
Batch size	64	16
# of hidden units	256	256
# of encoder layers	3	3
# of decoder layers	1	1
RNN cell	GRU	GRU
Optimizer	Adam	Adam
Input Dropout	0.8	1.0
Output Dropout	0.5	0.5

Table 1: The list of hyper-parameters tuned for both models.

Evaluation Metric	Model #1	Model #2
BLEU	0.8629	0.8611
NIST	8.2834	8.2004
METEOR	0.4569	0.4763
ROUGE_L	0.7159	0.7305
CIDEr	2.2774	2.3166

Table 2: The results of automatic evaluation on the validation set.

while the other one does not. The second model (*Model #2*) differs from the first one in the way the input sequence is created. It uses the concatenation of the de-lexicalized meaning representations (the same input as the first model takes) and the sequence of values of attributes of the meaning representations. Figure 3 shows an example of the input of the first and the second model.

We tune the hyper-parameters of the models based on the automatic evaluation metrics (i.e. BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Denkowski and Lavie, 2014), ROUGE_L (Lin, 2004) and CIDEr (Vedantam et al., 2015)). Table 1 shows the optimized values of the hyper-parameters for both models.

5 Results & Discussion

The parameter tuning helps us to choose the best model. We evaluated the trained models on the validation set to choose the best model configuration (see Table 1). Table 2 shows the results of the two models on the validation set.

For both models, we tried beam search decoder with various beam sizes. On the validation set, the beam search decoder shows no difference. On the test set, we used the beam search decoder with beam size of 5. The automatically generated sentences of the test set were evaluated automatically (using BLEU, NIST, METEOR, ROUGE_L and CIDEr) and by human annotators (Dušek et al., 2018). Table 3 shows the results of automatic evaluation of the test set. The manual evaluation is performed only for our primary submission, which is Model #2 with beam search. The reasons for selecting Model #2 as one of our primary submissions are: (1) according to Table 2, Model #2 outperforms Model #1 in 3 out of the 5 automatic metrics, (2) though Model #2 has a lower BLEU score compared to Model #1, this difference is not substantial, and (3) Model #2 uses the concatenated values as input and we were expecting this provide more information to the seq-to-seq model for better generations. The two metrics used for manual evaluation are *quality* and *naturalness*. In terms of *quality*, our submission ranked as third (in a scale of one to four) with the quality score of -0.169. Also, our primary submission achieves the *naturalness* score of -0.051, ranking in third place (in a scale of one to five).

The input sequence for Model #1	The input sequence for Model #2
name *name_x*, food *Indian*, priceRange *priceRange_x*, customer rating *customerRating_x*, area *city centre*, familyFriendly *yes*, near *near_x*.	name *name_x*, food *Indian*, priceRange *priceRange_x*, customer rating *customerRating_x*, area *city centre*, familyFriendly *yes*, near *near_x*. *name_x*, *Indian*, *priceRange_x*, *customerRating_x*, *city centre*, *yes*, *near_x*.

Figure 3: An example of the input of *Model #1* (left) and *Model #2* (right).

Evaluation Metric	Baseline	Model #1		Model #2	
		beam search	w/o beam search	beam search	w/o beam search
BLEU	0.6593	0.6201	0.6182	0.6336	0.6208
NIST	8.6094	8.0938	8.0616	8.1848	8.0632
METEOR	0.4483	0.4419	0.4417	0.4322	0.4417
ROUGE_L	0.6850	0.6740	0.6729	0.6828	0.6692
CIDEr	2.2338	2.1251	2.0783	2.1425	2.1127

Table 3: The results of automatic evaluation on the test set.

Although this proposed model is an end-to-end approach, there are some limitations that should be explored further. One of the limitations is that we do not have any control on the decoder to generate all the attributes that appeared in the meaning representations. As a result, the model may suffer from not generating all the attributes or generating extra attributes. In both cases, the re-lexicalization component either cannot re-lexicalize all the placeholders or there are extra placeholders that cannot be re-lexicalized. 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. In addition, this model needs to be trained on a larger training set. For future work, we plan to use the released data set for generating semantically similar sentences for the meaning representations.

References

D. Britz, A. Goldie, T. Luong, and Q. Le. 2017. Massive Exploration of Neural Machine Translation Architectures.

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* .

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.

Nina Dethlefs, Helen Wright Hastie, Heriberto Cuayáhuitl, and Oliver Lemon. 2013. Conditional Random Fields for Responsive Surface Realisation using Global Features. In *ACL (1)*. pages 1254–1263.

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.

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

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 *EMNLP*. volume 3, page 413.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research* 48:305–346.

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* .

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.

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.

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

Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. pages 1009–1018.

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 Ondrej Dušek, Jekaterina Novikova, and Verena Rieser, editors, *(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.

Marilyn A Walker, Owen Rambow, and Monica Rogati. 2001. SPoT: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*. pages 1–8.

Yuk Wah Wong and Raymond J Mooney. 2007. Generation by Inverting a Semantic Parser that Uses Statistical Machine Translation. In *HLT-NAACL*. pages 172–179.