Char2char Generation with Reranking for the E2E NLG Challenge

Shubham Agarwal
Heriot-Watt University [*]
Edinburgh, UK
sa201@hw.ac.uk

Marc Dymetman
NAVER Labs Europe[†]
Grenoble, France
marc.dymetman@naverlabs.com

Éric Gaussier
Université Grenoble Alpes
Grenoble, France
Eric.Gaussier@imag.fr

Abstract

This paper describes our submission to the E2E NLG Challenge. Recently, neural seq2seq approaches have become mainstream in NLG, often resorting to pre- (respectively post-) processing *delexicalization* (relexicalization) steps at the word-level to handle rare words. By contrast, we train a simple character level seq2seq model, which requires no pre/post-processing (delexicalization, tokenization or even lowercasing), with surprisingly good results. For further improvement, we explore two re-ranking approaches for scoring candidates. We also introduce a synthetic dataset creation procedure, which opens up a new way of creating artificial datasets for Natural Language Generation.

1 Introduction

Natural Language Generation from Dialogue Acts involves generating human understandable utterances from slot-value pairs in a Meaning Representation (MR). This is a component in Spoken Dialogue Systems, where recent advances in Deep Learning are stimulating interest towards using end-to-end models. Traditionally, the Natural Language Generation (NLG) component in Spoken Dialogue Systems has been rule-based, involving a two stage pipeline: 'sentence planning' (deciding the overall structure of the sentence) and 'surface realization' (which renders actual utterances using this structure). The resulting utterances using these rule-based systems tend to be rigid, repetitive and limited in scope. Recent approaches in dialogue generation tend to directly learn the utterances from data (Mei et al., 2015; Lampouras and Vlachos, 2016; Dušek and Jurčíček, 2016; Wen et al., 2015).

Recurrent Neural Networks with gated cell variants such as LSTMs and GRUs (Hochreiter and Schmidhuber, 1997; Cho et al., 2014) are now extensively used to model sequential data. This class of neural networks when integrated in a Sequence to Sequence (Cho et al., 2014; Sutskever et al., 2014) framework have produced state-of-art results in Machine Translation (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015), Conversational Modeling (Vinyals and Le, 2015), Semantic Parsing (Xiao et al., 2016) and Natural Language Generation (Wen et al., 2015; Mei et al., 2015). While these models were initially developed to be used at word level in NLP related tasks, there has been a recent interest to use character level sequences, as in Machine Translation (Chung et al., 2016; Zhao and Zhang, 2016; Ling et al., 2016).

Neural seq2seq approaches to Natural Language Generation (NLG) are typically word-based, and resort to delexicalization (a process in which named entities (slot values) are replaced with special 'placeholders' (Wen et al., 2015)) to handle rare or unknown words (out-of-vocabulary (OOV) words, even with a large vocabulary). It can be argued that this de-lexicalization is unable to account for phenomena such as morphological agreement (gender, numbers) in the generated text (Sharma et al., 2016; Nayak et al., 2017).

However, Goyal et al. (2016) and Agarwal and Dymetman (2017) employ a char-based seq2seq model where the input MR is simply represented as a character sequence, and the output is also generated char-by-char; avoiding the rare word problem, as the character vocabulary is very small.

[*] Work done during internship at Naver Labs (Previously *Xerox Research Centre Europe*.)

[†] Previously *Xerox Research Centre Europe*.

Proceedings of The 11th International Natural Language Generation Conference, pages 451–456,
Tilburg, The Netherlands, November 5-8, 2018. ©2018 Association for Computational Linguistics

This work builds on top of the formulation of Agarwal and Dymetman (2017) and describes our submission for the E2E NLG challenge (Novikova et al., 2017). We further explore re-ranking techniques in order to identify the perfect 'oracle prediction' utterance. One of the strategies for re-ranking uses an approach similar to the 'inverted generation' technique of (Chisholm et al., 2017). Sennrich et al. (2015), Li et al. (2015) and Konstas et al. (2017) have also trained a reverse model for back translation in Machine Translation and NLG. A synthetic data creation technique is used by Dušek et al. (2017) and Logacheva and Specia (2015) but as far as we know, our protocol is novel. Our contributions in this paper and challenge can, thus, be summarized as:

1. We show how a vanilla character-based sequence-to-sequence model performs successfully on the challenge test dataset in terms of BLEU score, while having a tendency to omit semantic material. As far as we know, we are the only team using character based seq2seq for the challenge.
2. We propose a novel data augmentation technique in Natural Language Generation (NLG) which consists of 'editing' the Meaning Representation (MR) and using the original ReFerences (RF). This fabricated dataset helps us in extracting features (to detect errors), used for re-ranking the generated candidates (Section 2.2).
3. We introduce two different re-ranking strategies corresponding to our primary and secondary submission (in the challenge), defined in Section 2.3.[1]

2 Model

In the sequel, we will refer to our vanilla char2char model with the term Forward Model.

2.1 Forward Model

We use a Character-based Sequence-to-Sequence RNN model (Sutskever et al., 2014; Cho et al., 2014) with attention mechanism (Bahdanau et al., 2015). We feed a sequence of embeddings of the individual characters composing the source Meaning Representation (MR) -seen as a string- to the Encoder RNN and try to predict the character sequence of the corresponding utterances (RF) in the generation stage with the Decoder RNN.

Coupled with the attention mechanism, seq2seq models have become de-facto standard in generation tasks. The encoder RNN embeds each of the source characters into vectors exploiting the hidden states computed by the RNN. The decoder RNN predicts the next character based on its current hidden state, previous character, and also the "context" vector c_i, computed by the attention model.

While several strategies have been proposed to improve results using Beam Search in Machine Translation (Freitag and Al-Onaizan, 2017), we used the length normalization (aka length penalty) approach Wu et al. (2016) for our task. A heuristically derived length penalty term is added to the scoring function which ranks the probable candidates used to generate the best prediction.

2.2 Protocol for synthetic dataset creation

We artificially create a training set for the classifier (defined in Section 2.3.2) to detect errors (primarily omission of content) in generated utterances, by a data augmentation technique. The systematic structure of the slots in MR gives us freedom to naturally augment data for our use case. To the best of our knowledge, this is the first approach of using data augmentation in the proposed fashion and opens up new directions to create artificial datasets for NLG. We first define the procedure for creating a dataset to detect omission and then show how a similar approach can be used to create a synthetic dataset to detect additions.

Detecting omissions. This approach assumes that originally there are no omissions in RF for a given MR (in the training dataset). These can be considered as positive pairs when detecting omissions. Now if we artificially add another slot to the original MR and use the same RF for this new (constructed) MR, naturally the original RF tends to show omission of this added slot.

$$\begin{aligned} MR_{original} &\xrightarrow{\text{+ Added slot}} MR_{new} \\ MR_{original} &\xrightarrow{\text{- Removed slot}} MR_{new} \end{aligned} \quad (1)$$

This is a two stage procedure: (a) Select a slot to add. (b) Select a corresponding slot value. Instead of sampling a particular slot in step (a), we add all the slots one by one (that could be augmented in MR apart from currently present slots).

[1] Due to space limitations, our description here omits a number of aspects. For a more extensive description, analysis and examples, please refer to http://www.macs.hw.ac.uk/InteractionLab/E2E/final_papers/E2E-NLE.pdf.

Having chosen the slot type to be added, we add the slot value according to probability distribution of the slot values for that slot type. The original ($\text{MR}_{original}$,RF) pair is assigned a class label of 1 and the new artificial pairs (MR_{new},RF) a label of 0, denoting a case of omission (first line of (1)). Thus, these triplets (MR, RF, Class Label) allow us to treat this as a classification task.

Detecting additions. In order to create a dataset which can be used for training our model to detect additions, we proceed in a similar way. The difference is that now we systematically remove one slot in the original MR to create the new MRs (second line of (1)).

In both cases, we control the procedure by manipulating MRs instead of the Natural Language RF. This kind of augmented dataset opens up the possibility of using any classifier to detect the above mentioned errors.

2.3 Re-ranking Models

In this section, we define two techniques to re-rank the n-best list and these serve as primary and secondary submissions to the challenge.

2.3.1 Reverse Model

We generated a list of top-k predictions (using Beam Search) for each MR in what we call the *forward* phase of the model. In parallel, we trained a *reverse* model which tries to reconstruct the MR given the target RF, similar to the autoencoder model by Chisholm et al. (2017). This is guided by an intuition that if our prediction omits some information, the reverse reconstruction of MR would also tend to omit slot-value pairs for the omitted slot values in the prediction. We then score and re-rank the top-k predictions based on a distance metric, namely the edit distance between the original MR and the MR generated by the reverse model, starting from the utterance predicted in the forward direction.

To avoid defining the weights when combining edit distance with the log probability of the model, we used a simplified mechanism. At the time of re-ranking, we choose the first output in our n-best list with zero edit distance as our prediction. If no such prediction can be found, we rely upon the first prediction in our (probabilistically) sorted n-best list. Figure 1 illustrates our pipeline approach.

2.3.2 Classifier as a re-ranker

To treat omission (or more generally any kind of *semantic adequacy* mis-representation such as repetition or addition of content) in the predictions as a classification task, we developed a dataset (consisting of triplets) using the protocol defined earlier. However, to train the classifier we relied on hand-crafted features based on string matching in the prediction (with corresponding slot value in the MR). In total, there were 7 features, corresponding to each slot (except 'name' slot). To maintain the class balance, we replicated the original (MR,RF) pair (with a class label of 1) for each artificially generated (MR,RF) pair (with a class label of 0, corresponding to omissions).

We used a logistic regression classifier to detect omissions following a similar re-ranking strategy as defined for the reverse model. For each probable candidate by the forward model, we first extracted these features and predicted the label by this logistic regression classifier. The first output in our n-best list with a class label 1 is then chosen as the resulting utterance. As a fallback mechanism, we rely on the best prediction by the forward model (similar to the reverse model). We chose the primary submission to the challenge as the pipeline model with classifier as re-ranker. Our second submission was based on re-ranking using the reverse model while the vanilla forward char2char model was our third submission.

3 Experiments

The updated challenge dataset comprises 50K canonically ordered and systematically structured (MR,RF) pairs, collected following the crowdsourcing protocol explained in Novikova et al. (2016). Consisting of 8 different slots (and their respective different values), note that the statistics in the test set differ significantly from the training set. We used the open source *tf-seq2seq* framework[2], built over TensorFlow (Abadi et al., 2016) and provided along with (Britz et al., 2017), with some standard configurations. We experimented with different numbers of layers in the encoder and decoder as well as different beam widths, while using the bi-directional encoder with an "additive" attention mechanism. In terms of BLEU, our best performing model had the following configuration: encoder 1 layer, decoder 2 layers, GRU cell, beam-width 20, length penalty 1.

[2] https://github.com/google/seq2seq.

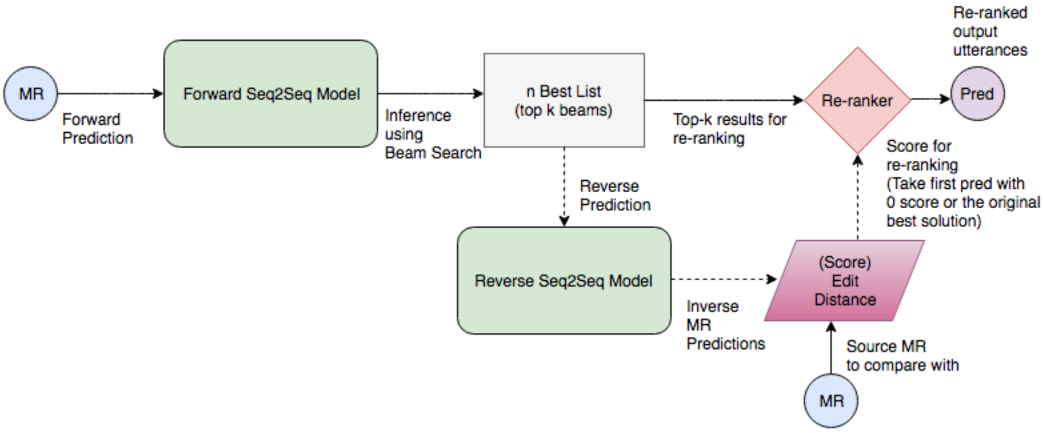
Re-ranked output utterances
MR
Forward Prediction
Forward Seq2Seq Model
Inference using Beam Search
n Best List (top k beams)
Top-k results for re-ranking
Re-ranker
Pred
Score for re-ranking (Take first pred with 0 score or the original best solution)
Reverse Prediction
Reverse Seq2Seq Model
Inverse MR Predictions
(Score) Edit Distance
Source MR to compare with
MR

Figure 1: Illustration of the pipeline for the re-ranking approach (based on inverse reconstructions using reverse model) as described in Section 2.3. Apart from Forward and Reverse seq2seq models, we have a re-ranker based on the edit distance of the actual MR and the inverse reconstructed MR.

4 Evaluation

We chose our primary system to be the re-ranker using the classifier. Table 1 summarizes our ranking among all the 60+ submissions (primary as well as additional) on the test set. In terms of BLEU, two of our systems were in the top 5 among all 60+ submissions to the challenge.

Submission	BLEU	Overall Rank
Re-ranking using classifier (Primary)	0.653	18
Re-ranking using reverse (Secondary)	0.666	5
Forward (Third)	0.667	4
Baseline	0.659	10

Table 1: Automatic BLEU evaluations released by organizers on the final challenge submission. We had 3 submissions as described in Section 2. Two of our systems were in the top 5 among all 60+ submissions.

Metric	TrueSkill	Range	Cluster
Quality	0.048	(8-12)	2
Naturalness	0.105	(4-8)	2

Table 2: Human evaluation was crowd-sourced on the primary system according to the TrueSkill algorithm (Sakaguchi et al., 2014)

Results for human evaluation, as released by the challenge organizers, are summarized in Table 2 of (Dušek et al., 2018). They followed the TrueSkill algorithm (Sakaguchi et al., 2014) judging all the primary systems on *Quality* and *Naturalness*. We obtained competitive results in terms of both metrics, our system being in the 2nd cluster out of 5 (for both evaluations). On the other hand, most systems ranked high on quality tended to have lower ranks for naturalness and vice versa.

5 Analysis

We found that the presence of an ‘oracle prediction’ (perfect utterance) was dependent on the number of slots in the MR. When the number of slots was 7 or 8, the presence of an oracle in the top-20 predictions decreased significantly, as opposed to the case when the number of slots was less than 7. However, the most prominent issue was that of omissions, among the utterances produced in first position (by forward model). There were no additions or non-words. We observed a similar issue of omissions in human references (target for our model) as well. Our two different strategies, thus, improved the semantic adequacy by re-ranking the probable candidates and successfully finding the ‘oracle’ prediction in the top-20 list. However, in terms of automatic evaluation, the BLEU score showed an inverse relationship with adequacy. Nevertheless, we chose our primary system to be the re-ranker with a classifier over the forward model.

We did not find any issues while “copying” the restaurant ‘name’ or ‘near’ slots on the dev set. However, on the test set, as the statistics of the data changed in terms of both slots, we found a tendency of the model to generate the more frequent slot values (corresponding to both slots in the training dataset), instead of copying the actual slot value.

6 Conclusion

We show how a char2char model can be employed for the task of NLG and show competitive results in this challenge. Our vanilla character based model, building on Agarwal and Dymetman (2017), requires minimal effort in terms of any processing of dataset while also producing great diversity in the generated utterances. We then propose two re-ranking strategies for further improvements. Even though re-ranking methods show improvements in terms of semantic adequacy, we find a reversal of trend in terms of BLEU.

Our synthetic data creation technique could be adapted for augmenting NLG datasets and the classifier-based score could also be used as a reward in a Reinforcement Learning paradigm.

Acknowledgements

We like to thank Chunyang Xiao and Matthias Gallé for their useful suggestions and comments.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR abs/1603.04467*.

Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the E2E NLG challenge dataset. In *Proc. SIGdial*. ACL.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *CoRR abs/1703.03906*.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from wikidata. *CoRR abs/1702.06235*.

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. In *Proc. EMNLP*.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *CoRR abs/1603.06147*.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *CoRR abs/1606.05491*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2017. Referenceless quality estimation for natural language generation. *arXiv preprint arXiv:1708.01759*.

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

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *CoRR abs/1702.01806*.

Raghav Goyal, Marc Dymetman, and Eric Gaussier. 2016. Natural Language Generation through Character-based RNNs with Finite-State Prior Knowledge. In *Proc. COLING*, Osaka, Japan.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. *CoRR abs/1704.08381*.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proc. COLING*, pages 1101–1112.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR abs/1510.03055*.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2016. Character-based neural machine translation. In *Proc. ICLR*, pages 1–11.

Varvara Logacheva and Lucia Specia. 2015. The role of artificially generated negative data for quality estimation of machine translation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. *CoRR abs/1509.00838*.

Neha Nayak, Dilek Hakkani-Tur, Marilyn Walker, and Larry Heck. 2017. To plan or not to plan? sequence to sequence generation for language generation in dialogue systems.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proc. SIGdial, ACL*, Saarbrücken, Germany.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing NLG Data: Pictures Elicit Better Data. *CoRR abs/1608.00339*.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proc. Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA. ACL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR abs/1511.06709*.

Shikhar Sharma, Jing He, Kaheer Suleman, Hannes Schulz, and Philip Bachman. 2016. Natural language generation in dialogue using lexicalized and delexicalized data. *arXiv preprint arXiv:1606.03632*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*, pages 3104–3112.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *CoRR abs/1506.05869*.

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. In *Proc. EMNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klinger, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz 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*.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. *Proc. ACL*.

Shenjian Zhao and Zhihua Zhang. 2016. An efficient character-level neural machine translation. *CoRR abs/1608.04738*.