The Unbearable Weight of Generating Artificial Errors for Grammatical Error Correction

Phu Mon Htut*
Center for Data Science
New York University
pmh330@nyu.edu

Joel Tetreault
Grammarly
joel.tetreault@grammarly.com

Abstract

In recent years, sequence-to-sequence models have been very effective for end-to-end grammatical error correction (GEC). As creating human-annotated parallel corpus for GEC is expensive and time-consuming, there has been work on artificial corpus generation with the aim of creating sentences that contain realistic grammatical errors from grammatically correct sentences. In this paper, we investigate the impact of using recent neural models for generating errors to help neural models to correct errors. We conduct a battery of experiments on the effect of data size, models, and comparison with a rule-based approach.

1 Introduction

Grammatical error correction (GEC) is the task of automatically identifying and correcting the grammatical errors in the written text. Recent work treats GEC as a translation task that use sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015) to rewrite sentences with grammatical errors to grammatically correct sentences. As with machine translation models, GEC models benefit largely from the amount of parallel training data. Since it is expensive and time-consuming to create annotated parallel corpus for training, there is research into generating sentences with artificial errors from grammatically correct sentences with the goal of simulating human-annotated data in a cost-effective way (Yuan and Briscoe, 2016; Xie et al., 2016; Chollampatt and Ng, 2018).

Recent work in artificial error generation (AEG) is inspired by the back-translation approach of machine translation systems (Sennrich et al., 2016; Poncelas et al., 2018). In this framework, an intermediate model is trained to translate correct sentences into errorful sentences. A new parallel corpus is created using the largely available grammatically correct sentences and the corresponding synthetic data generated by this intermediate model. The newly created corpus with the artificial errors is then used to train a GEC model (Rei et al., 2017; Xie et al., 2018; Ge et al., 2018).

To date, there is no work that compares how different base model architectures perform in the AEG task. In this paper, we investigate how effective are different model architectures in generating artificial, parallel data to improve a GEC model. Specifically, we train four recent neural models (and one rule-based model (Bryant and Briscoe, 2018)), including two new syntax-based models, for generating as well as correcting errors. We analyze which models are effective in the AEG and correction conditions as well as by data size. Essentially, we seek to understand how effective are recent sequence-to-sequence (seq2seq) neural model as AEG mechanisms "out of the box."

2 Related Work

Before the adoption of neural models, early approaches to AEG involved identifying error statistics and patterns in the corpus and applying them to grammatically correct sentences (Brockett et al., 2006; Rozovskaya and Roth, 2010). Inspired by the back-translation approach, recent AEG approaches inject errors into grammatically correct input sentences by adopting methods from neural machine translation (Felice and Yuan, 2014; Kasewa et al., 2018). Xie et al. (2018) propose an approach that adds noise to the beam-search phase of an back-translation based AEG model to generate more diverse errors. They use the synthesized parallel data generated by this method to train a multi-layer convolutional GEC model and achieve a 5 point $F_{0.5}$ improvement on the CoNLL-2014 test data (Ng et al., 2014).

*Work done during internship at Grammarly

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 478–483
Florence, Italy, August 2, 2019.

Ge et al. (2018) propose a fluency-boosting learning method that generates less fluent sentences from correct sentences and pairs them with correct sentences to create new error-correct sentence pairs during training. Their GEC model trained with artificial errors approaches human-level performance on multiple test sets.

3 Approach

3.1 Correction and Generation Tasks

We train our models on the two tasks—error correction and error generation. In *error correction*, the encoder of the sequence-to-sequence model takes an errorful sentence as input and the decoder outputs the grammatically correct sentence. The process is reversed in the *error generation* task, where the model takes a correct sentence as input and produces an errorful sentence as the output of the decoder.

We investigate four recent neural sequence-to-sequence models—(i) multi-layer convolutional model (MLCONV; Chollampatt and Ng, 2018), (ii) Transformer (Vaswani et al., 2017), (iii) Parsing-Reading-Predict Networks (PRPN; Shen et al., 2018), (iv) Ordered Neurons (ON-LSTM; Shen et al., 2019)—as error correction models as well as error generation models. The PRPN and ON-LSTM models are originally designed as recurrent language models that jointly learn to induce latent constituency parse trees. We use the adaption of PRPN and ON-LSTM models as decoders of machine translation systems (UnderReview, 2019): In this setting, a 2-layer LSTM is used as the encoder of the syntactic seq-to-seq models, and the PRPN and ON-LSTM are implemented as the decoders with attention (Bahdanau et al., 2015). We hypothesize that syntax is important in GEC and explore whether models that incorporate syntactic bias would help with GEC task. We provide a brief description of each model in §3.2 and refer readers to the original work for more details.

3.2 Models

Multi-layer Convolutional Model We use the multi-layer convolutional encoder-decoder base model (MLCONV) of Chollampatt and Ng (2018) using the publicly available code from the authors.[1] As our aim is to only compare the performance of different architectures and not to achieve state-of-the-art performance, we make few changes to their code. The model of Chollampatt and Ng (2018) produces 12 possible correct sentences for each input sentences with error. They also train an N-gram language model as a reranker to score the generated sentences and pick the corrected sentence with the best score as final output. We did not use this re-ranking step in our model, nor did we perform ensembling or use the pre-trained embeddings as in the original work. We do not observe improvement in models like transformer and PRPN using re-ranking with an N-gram language model. Additionally, there's only a slight improvement in MLCONV using re-ranking. The reason might be because the N-gram language model is not very powerful.

Transformer Model We use the publicly available Fairseq framework which is built using Pytorch for training the Transformer model. We apply the same hyper-parameters used for training the IWSLT'14 German-English translation model in the experiments of Vaswani et al. (2017).

PRPN Model is a language model that jointly learns to parse and perform language modeling (Shen et al., 2018). It uses a recurrent module with a self-attention gating mechanism and the gate values are used to construct the constituency tree. We use the BiLSTM model as the encoder and PRPN as the decoder of the sequence-to-sequence model.

ON-LSTM Model is follow-up work of PRPN, which incorporates syntax-based inductive bias to the LSTM unit by imposing hierarchical update order on the hidden state neurons (Shen et al., 2019). ON-LSTM assumes that different nodes of a constituency trees are represented by the different chunks of adjacent neurons in the hidden state, and introduces a master forget gate and a master input gate to dynamically allocate the chunks of hidden state neurons to different nodes. We use a BiLSTM model as encoder and ON-LSTM model as decoder.

4 Experiments

4.1 Data

We use the NUS Corpus of Learner English (NUCLE; Dahlmeier et al., 2013) and the Cambridge Learner Corpus (CLC; Nicholls, 2003) as base data for training both the correction and generation models. We remove sentence pairs that do

[1] https://github.com/nusnlp/mlconvgec2018

not contain errors during preprocessing resulting in 51,693 sentence pairs from NUCLE and 1.09 million sentence pairs from the CLC . We append the CLC data to the NUCLE training set (henceforth NUCLE-CLC) to use as training data for both AEG and correction. We use the standard NUCLE development data as our validation set and we early-stop the training based on the cross-entropy loss of the seq-to-seq models for all models. For the generation of synthetic errorful data, we use the 2017 subsection of the LDC New York Times corpus also employed in the error generation experiments of Xie et al. (2018) which contains around 1 million sentences.[2]

4.2 Setup

We conduct four experiments in this paper. In **Exp1**, we train all the AEG models and intermediate GEC models on NUCLE-CLC. We use the NYT dataset as input to the AEG models to generate sentences with artificial errors. We then create new parallel training sets for correction by combining the sentences from CLC and NUCLE with the errorful sentences generated by each model. We then train the GEC models using these parallel datasets.

The three other experiments are variants of the first. In **Exp2** we train all correction models on artificial errors generated by the top neural AEG systems and a rule-based system for comparison. In **Exp3**, we train the GEC models on NUCLE to analyze models built on real data. Finally, in **Exp4**, we train all GEC models on artificial data to determine how well correction models can perform without any real data.

All our experiments are tested on the CoNLL-2014 test set and we use the sentence-level $F0.5$ score from the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) for evaluation. All models are implemented using the Fairseq framework.[3]

4.3 Results

Exp1: Figure 1 shows the performance of GEC models trained on the base NUCLE-CLC set and then retraining with various amounts of artificial data. We first observe that PRPN performs substantially higher than the rest of the models when trained only with the base CLC-NUCLE data. However, its performance drops when artificial data generated by the corresponding PRPN AEG model is added. As for ON-LSTM, the performance improves slightly when the amount of added data is less than 100k but the performance drops drastically otherwise. Conversely, the performance of MLCONV and Transformer improves with the added artificial data but the improvement is not linear with the amount of added data.

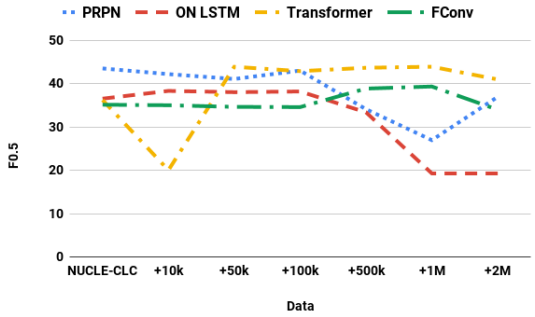

Figure 1: (Exp1) Models trained on the artificial data generated by the corresponding AEG model. The X-axis represents the amount of artificial data added to NUCLE-CLC during training.

Exp2: Since the performance of MLCONV and Transformer GEC models improve with the addition of artificial data generated by corresponding AEG models, we hypothesize that the artificial error generated by these models are useful. To test this hypothesis, we train all the GEC models with various amount of artificial error generated by MLCONV and Transformer AEG models. We also compare these AEG models to a rule-based one inspired by the confusion set generation method in Bryant and Briscoe (2018). We subsequently score each sentence with a language model (GPT-2 (Radford et al., 2018)) in order not to select the most probable sentence. This method generates a confusion set for prepositions (set of prepositions plus an empty element), determiners, and morphological alternatives (cat → cats).

The results of these experiments are found in Table 1. Nearly all correction models improve when using MLCONV or Transformer AEG data with the biggest performances yielded using the Transformer model. Interestingly, when using 1M or 2M samples, performance starts to decline. We believe that over 1M samples, the noisiness of the artificial data overwhelms the contributions of the real data (roughly over 1M samples). The performance of all models drops when trained with

[2] https://catalog.ldc.upenn.edu/LDC2008T19

[3] https://github.com/pytorch/fairseq

GEC Model	AEG model	NUCLE-CLC	10K	50K	100K	500K	1M	2M
MLCONV	MLCONV	35.2	35.1	34.7	34.6	38.9	39.4	34.0
Transformer	MLCONV	36.3	43.9	**44.1**	**45.4**	**44.4**	**45.5**	**42.0**
PRPN	MLCONV	**43.6**	**45.4**	42.8	43.2	39.6	38.6	31.7
ON-LSTM	MLCONV	36.6	39.8	35.6	38.4	36.9	24.2	20.1
MLCONV	Transformer	35.2	36.1	35.2	39.4	36.6	36.6	36.1
Transformer	Transformer	36.3	20.1	**43.9**	**42.9**	**43.7**	**44.0**	**41.0**
PRPN	Transformer	**43.6**	**43.1**	40.9	40.6	41.4	29.4	31.7
ON-LSTM	Transformer	36.6	39.8	38.2	39.6	24.0	21.3	20.1
MLCONV	Rule-based	35.2	6.0	7.8	10.5	13.7	13.9	–
Transformer	Rule-based	36.3	**13.5**	**14.4**	**21.8**	**14.5**	**21.6**	–
PRPN	Rule-based	**43.6**	2.8	4.9	2.6	3.9	8.9	–
ON-LSTM	Rule-based	36.6	4.7	3.9	5.5	4.2	5.3	–

Table 1: (Exp2) Evaluating the impact of MLCONV, Transformer and the rule-based AEG systems. NUCLE-CLC column represents the F0.5 score of GEC models trained on the base NUCLE-CLC data. *10K, 50K, 100K, 500K, 1M, and 2M* represents the amount of artificial data added to the NUCLE-CLC during training.

the errors generated by the rule-based model. It is interesting to observe that the performance drops significantly just by adding 10K artificial sentences to the base data.

Exp3: Table 2 shows the performance of the models trained on NUCLE dataset with additional artificial data generated by corresponding AEG models trained on NUCLE-CLC. We can see that the performance of all models, except ON-LSTM, improves significantly when 1 million artificial sentence pairs are added to the NUCLE training data, even though the errors in these sentences do not resemble natural errors. This contrasts with the result in Figure 1 where the performance of the GEC models trained with the combination of artificial error and CLC-NUCLE base data drops. This suggests that artificial data is helpful when the base data size is relatively small.

Model	NUCLE	+10K	+50K	+1M
MLCONV	10.1	12.3	12.9	16.1
Transformer	11.2	**28.1**	**16.9**	22.8
PRPN	8.3	6.9	12.5	**26.2**
ON-LSTM	9.4	11.3	11.8	6.0

Table 2: (Exp3) Using only NUCLE as base training for correction. The AEG models are trained using NUCLE-CLC data as in other experiments.

Exp4: The GEC models trained only on artificial data perform very poorly. The best setups, Transformer and MLCONV, achieve F0.5 scores of 12.8 and 12.4 respectively when trained with 2 million sentences generated by the corresponding AEG model. This outcomes suggests that AEG data should be paired with some sample of real data to be effective.

4.4 Manual Evaluation

We performed a manual analysis of the generated error sentences and found that many of the errors did not always resemble those produced by humans. For example, *The situation with other types is not much (better → downward)*. This shows that despite the noisiness of the error-generated data, some models (namely MLCONV and Transformer) were robust enough to improve. This also suggests that we may achieve better improvement by controlling artificial errors to resemble the errors produced by humans. The performance of syntax-based models goes down significantly with the addition of artificial errors, which indicates that these models may be sensitive to poor artificial data.

5 Conclusion

We investigated the potential of recent neural architectures, as well as rule-based one, to generate parallel data to improve neural GEC. We found that the Multi-Layer Convolutional and Transformer models tended to produce data that could improve several models, though too much of it would begin to dampen performance. The most substantial improvements could be found when the size of the real data for training was quite small. We also found that the syntax-based models, PRPN and ONLSTM, are very sensitive to the quality of artificial errors and their performance drops substantially with the addition of artificial error data. Our experiments suggest that, unlike in machine translation, it is not very straightforward

to use a simple back-translation approach for GEC as unrealistic errors produced by back-translation can hurt the correction performance substantially.

We believe this work shows the promise of using recent neural methods in an out-of-the-box framework, though with care. Future work will focus on ways of improving the quality of the synthetic data. Ideas include leveraging recent developments in powerful language models or better controlling for diversity and frequency of specific error types.

Acknowledgements

We would like to thank the Grammarly Research Team, especially Maria Nadejde, Courtney Napoles, Dimitris Alikaniotis, Andrey Gryschuck, Maksym Bezva and Oleksiy Syvokon. We would also like to thank Sam Bowman, Kyunghyun Cho, and the three anonymous reviewers for their helpful discussion and feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*.

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *BEA@NAACL-HLT*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS corpus of learner english. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*, pages 22–31.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 116–126.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *CoRR*, abs/1807.01270.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4977–4983.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Diane Nicholls. 2003. The cambridge learner corpus - error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. *CoRR*, abs/1804.06189.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 287–292.

Alla Rozovskaya and Dan Roth. 2010. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 154–162.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the*

54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers.

Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered Neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

UnderReview. 2019. Unknown title.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*, abs/1603.09727.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y. Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 619–628.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 380–386.