Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection

Sudhanshu Kasewa and **Pontus Stenetorp** and **Sebastian Riedel**
{sudhanshu.kasewa.16, p.stenetorp, s.riedel}@ucl.ac.uk
University College London

Abstract

Grammatical error correction, like other machine learning tasks, greatly benefits from large quantities of high quality training data, which is typically expensive to produce. While writing a program to automatically generate realistic grammatical errors would be difficult, one could learn the distribution of naturally-occurring errors and attempt to introduce them into other datasets. Initial work on inducing errors in this way using statistical machine translation has shown promise; we investigate cheaply constructing synthetic samples, given a small corpus of human-annotated data, using an off-the-rack attentive sequence-to-sequence model and a straight-forward post-processing procedure. Our approach yields error-filled artificial data that helps a vanilla bi-directional LSTM to outperform the previous state of the art at grammatical error detection, and a previously introduced model to gain further improvements of over 5% $F_{0.5}$ score. When attempting to determine if a given sentence is synthetic, a human annotator at best achieves 39.39 F_1 score, indicating that our model generates mostly human-like instances.

1 Introduction

There is an ever-growing number of people learning English as a second language; providing them with quick feedback to facilitate their learning is a crucial, labour-intensive endeavour. Part of this process is identifying and correcting grammatical errors, and several computational techniques have been developed to automate it (Rozovskaya and Roth, 2014; Junczys-Dowmunt and Grundkiewicz, 2016). For example, given an erroneous sentence *"I wanted to **goes** to the beach"*, the grammatical error correction task is to output the valid sentence *"I wanted to **go** to the beach"*. The task can be cast as a two-stage process, detection and correction, which can either be per-formed sequentially (Yannakoudakis et al., 2017), or jointly (Napoles and Callison-Burch, 2017).

Automated error correction performance is arguably still too low for practical consideration, perhaps limited by the amount of training data (Rei et al., 2017). High quality annotations are expensive to procure, and foreign language learners and commercial entities may feel uncomfortable granting access to their data. Instead, one could attempt to supplement existing manual annotations with synthetic instances. Such artificial samples are beneficial only when they share structure with the true distribution from which human errors are generated. Generative Adversarial Networks (Goodfellow et al., 2014) could be used for this purpose, but they are difficult to train, and require a large collection of sentences that are incorrect. One might attempt self-training (McClosky et al., 2006), where new instances are generated by applying a trained model to unannotated data, using high-confidence predictions as ground truth labels. However, in such a scheme, the expectation is that the unlabelled text already contains errors, which is not usually the case for most freely available text such as Wikipedia articles as they strive towards correctness.

In place of using machine translation (MT) to *correct* grammatical mistakes (Yuan and Felice, 2013; Junczys-Dowmunt and Grundkiewicz, 2014; Yuan and Briscoe, 2016), one might consider swapping the input and output streams, and instead learn to *induce errors* into error-free text, for the purpose of creating a synthetic training dataset (Felice and Yuan, 2014). Recently, Rei et al. (2017) used a statistical MT (SMT) system to induce errors into error-free text. Building on this work, and leveraging recent advances in neural MT (NMT), we used an off-the-shelf attentive sequence-to-sequence model (Britz et al., 2017), eliminating the need of specialised soft-

ware such as a phrase-table generator, decoder, and part-of-speech tagger. We created multiple synthetic datasets from in-domain and out-of-domain sources, and found that stochastic token sampling, and pruning redundant and low-likelihood sentences, were helpful in generating meaningful corruptions. Using the artificial samples thus generated, we improved upon detection results with simply a vanilla bi-directional LSTM (Hochreiter and Schmidhuber, 1997). Using a more powerful model, we established new state-of-the-art results, that improve on previously published $F_{0.5}$ scores by over 5%. Additionally, we confirm that our generated instances are human-like, as an annotator identifying generated sentences achieved a maximum F_1 score of 39.39.

2 Related work

In computer vision, images are blurred, rotated, or otherwise deformed inexpensively to create new training instances (Wang and Perez, 2017), because such manipulation does not significantly alter the image semantics. Similar coarse processes do not work in NLP since mutating even a single letter or a word can change a sentence's meaning, or render it nonsensical. Nonetheless, Vinyals et al. (2015) employed a kind of self-training where they use noisy predictions for unlabelled instances output by existing state-of-the-art parsers as ground-truth labels, and improved syntactic parsing performance. Sennrich et al. (2016) synthesised training instances by round-trip-translating a monolingual corpus with weaker versions of an NMT learner, and used them to improve the translation. Bouchard et al. (2016) developed an efficient algorithm to blend generated and true data for improving generalisation.

Grammar correction is a well-studied task in NLP, and early systems were rule-based pattern recognisers (Macdonald, 1983) and dictionary-based linguistic analysis engines (Richardson and Braden-Harder, 1988). Later systems used statistical approaches, addressing specific kinds of errors such as article insertion (Knight et al., 1994) and spelling correction (Golding and Roth, 1996). Most recently, architectural innovations in neural sequence labelling (Rei et al., 2016; Rei, 2017) raised error detection performance through improved ability to process unknown words and jointly learning a language model.

Early efforts for artificial error generation in-

cluded generating specific types of errors, such as mass noun errors (Brockett et al., 2006) and article errors (Rozovskaya and Roth, 2010), and leveraging linguistic information to identify error patterns and transfer them onto grammatically correct text (Foster and Andersen, 2009; Yuan and Felice, 2013). Imamura et al. (2012) investigated methods to generate pseudo-erroneous sentences for error correction in Japanese. Recently, Rei et al. (2017) corrupted error-free text using SMT to create training instances for error detection.

3 Neural error generation

To learn to introduce errors, we use an off-the-shelf attentive sequence-to-sequence neural network (Bahdanau et al., 2014). Given an input sequence, the *encoder* generates context vectors for each token. Then, the *attention mechanism* and the *decoder* work in tandem to emit a distribution over the target vocabulary. At every decoder time-step, the encoder context vectors are scored by the attention mechanism, and a weighted sum is supplied to the decoder, along with its propagated internal state and last output symbol.

Corruption: Tokens from this distribution are sampled at every decoder time-step, either by $argmax$ (AM), which emits the most likely word, or by a stochastic alternative such as *temperature sampling* (TS) as $argmax$ cannot be relied on to generate rare words. A temperature parameter $\tau > 0$ sharpens or softens the distribution:

$$\tilde{p}_i = \boldsymbol{f}_\tau(p)_i = \frac{p_i^{\frac{1}{\tau}}}{\sum_j p_j^{\frac{1}{\tau}}}$$

where i are the components of the probability distribution corresponding to words in the vocabulary. As one interpolates τ from 0 to 1, the behaviour of $\tilde{p}$ transitions from *argmax* to p, controlling the diversity of the generated tokens.

The sentence generated by TS might be a low probability sequence from the joint conditional distribution $P(\mathbf{v}|\mathbf{u})$, where $\mathbf{u}$ is the input sentence and $\mathbf{v}$ is the output sentence. One way around this is to use *beam search* (BS), which checks the likelihood of every possible continuation of a sentence fragment, and maintains a list of the n best translations generated up to the current time-step. AM, TS, and BS are indicative of the trade-off between increasing levels of model flexibility at the cost of computation; we compare them to assess whether

Original	Corruption
She promised to turn over a new leaf.	She promissed to turn over a new leaf.
At the moment I'm in Spain.	During the moment I'm in Spain.

Table 1: Example sentences generated by our NMT pipeline.

Data augmentation strategy	Model	FCE (dev)	FCE	CoNLL1	CoNLL2
Rei et al. (2017) FCE_{PAT} + EVP_{PAT}	SL	–	47.8	19.5	28.5
Rei et al. (2017) FCE_{SMT} + EVP_{SMT}	SL	–	48.4	19.7	28.4
Rei et al. (2017) $FCE_{SMT+PAT}$ + $EVP_{SMT+PAT}$	SL	–	49.1	21.9	30.1
None	BiLSTM	47.9	43.6	16.6	24.3
FCE_{TS}	BiLSTM	51.2	47.1	19.7	28.9
EVP_{BS}	BiLSTM	52.1	50.1	20.8	29.0
SW_{TS}	BiLSTM	51.5	50.6	24.2	31.7
FCE_{AM+TS}+EVP_{AM+TS}	BiLSTM	52.3	50.4	22.1	30.8
None	SL	52.5	48.2	17.4	25.5
FCE_{TS}	SL	54.8	49.9	20.9	29.2
EVP_{BS}	SL	55.2	54.6	23.3	31.4
SW_{TS}	SL	53.8	52.7	26.8	34.3
FCE_{AM+TS}+EVP_{AM+TS}	SL	**56.9**	54.6	25.1	33.0
FCE_{AM+TS}+EVP_{AM+TS}+SW_{AM+TS}	SL	56.5	**55.6**	**28.3**	**35.5**

Table 2: $F_{0.5}$ scores on various tests contrasted with published results and unaugmented baseline models.

the additional computations were helpful in creating high-quality synthetic instances.

Post-processing: Original and corrupted sentences are aligned at a word-level using Levenshtein distance. Using the minimal alignment, words in the corrupted sentence are labelled *correct*, 'c', or *incorrect*, 'i', as follows:

If the word is not aligned with itself, then 'i'. *Else*, if following a gap, then 'i', as at this point a human reader would notice that there is a word missing in the sentence. *Else*, if it is the last word, but it is not aligned to the last word of the source sentence, then 'i', as a human would realise that this sentence ends abruptly, *Else*, 'c'.

These token-labelled corrupted sentences now form an artificial dataset for training an error detector. Duplicate instances and corrupted sentences with more than 5 errors were dropped to remove noise from the downstream training.

4 Experiments

We evaluated our approach on the First Certificate of English (FCE) error detection dataset (Rei and Yannakoudakis, 2016), as well as on two human-annotated test sets (CoNLL1, CoNLL2) from the CoNLL 2014 shared task (Ng et al., 2014). The CoNLL data sets pose a unique challenge; as they are different in style and domain from FCE, we

have no matching training data. We compared the effect of different neural generation procedures (AM, TS, BS) and contrasted the downstream performance of a bidirectional LSTM with an elaborate sequence labeller.

4.1 Implementation details

NMT training and corruption: We minimally modified the open source implementation[1] of Britz et al. (2017) to implement TS and BS.[2] We trained our NMT with a single-layered encoder and decoder with cell size 256, on the parallel corpus version of FCE (Yannakoudakis et al., 2011), with early stopping after the FCE development set score dropped consistently for 20 epochs. We introduced errors into three datasets: FCE itself (450K tokens), the English Vocabulary Profile or EVP (270K tokens) and a subset of Simple Wikipedia or SW (8.4M tokens); of these, FCE and EVP were both used in artificial error generation via SMT and pattern extraction (PAT) by Rei et al. (2017), enabling us to make a fair experimental comparison. Ten corrupted versions using each of AM, TS ($\tau = 0.05$) and BS were sampled for FCE and EVP corruptions, while one sufficed for SW. The theoretical time complexity of BS is O(bn) for each sentence, where b is num-
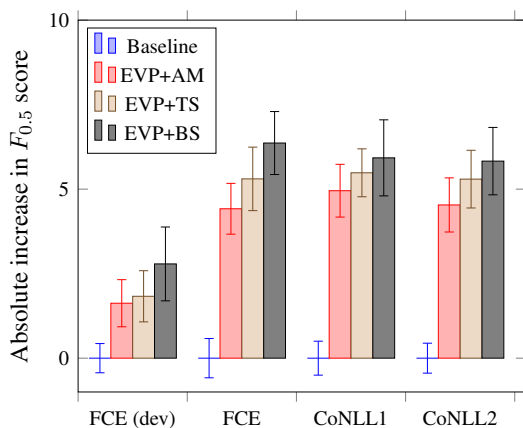
---
[1] https://github.com/google/seq2seq
[2] https://github.com/skasewa/wronging

Figure 1: Improvements using three different methods of generation.

Figure 2: Training with increasing amounts of corrupted data from FCE and SW.

ber of candidates, and n is the maximum length of a sentence. Empirically, BS with $b = 11$ took a factor of 11.3 more time than AM. Examples of generated errors are provided in Table 1.

Error detection: We compare two error detection models: a vanilla bi-directional LSTM (BiL-STM) (Schuster and Paliwal, 1997), and the state-of-the-art sequence labeller (SL) neural network used by Rei et al. (2017). These models were trained on the binary-labelled FCE training set augmented with the corrupted instances. Wherever no model is explicitly stated, the SL model was used. During training, we alternate between the annotated FCE dataset and the synthetic collection. This alternating protocol prevents overfitting on FCE; once it shifts back, it reinforces connections made from the helpful synthetic corruptions while forgetting about the noisy ones.

4.2 Results

The results for our baselines and data augmentation strategies can be found in Table 2. Augmented with our NMT generated data, even our vanilla downstream BiLSTM outperforms the SMT+PAT artificial error augmentation approach of Rei et al. (2017), indicating that our process better generalises the error information in the source dataset. Using the more powerful SL network bests the previous state of the art by over 5% on the FCE test. Most intriguingly, we note a significant improvement for the CoNLL tests using corruptions from out-of-domain SW. Figure 2 illustrates how we gain performance on these tests with increasing amounts of corrupted SW, which does not hold true for corrupted FCE. This shows that we were
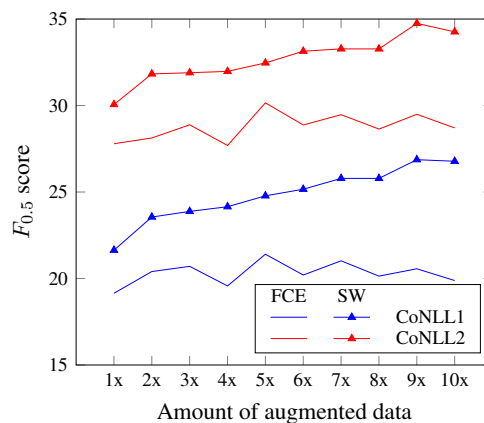
able to induce useful errors into a corpus with a large unseen vocabulary and different syntactic biases, and this in turn proved valuable for detecting errors in a third domain, suggesting that our method can transfer learned distributions across stylistic genres.

Using EVP as a standard source, Figure 1 illustrates the variance of the different sampling methods. All generation methods yield corruptions that significantly improve test performance, with instances sampled by beam-search consistently outperforming the alternatives.

5 Discussion

5.1 Error distribution

The original FCE dataset was annotated using the error taxonomy specified in Nicholls (2003), and contains 75 unique error codes. We annotated samples of EVP corrupted by all three sampling methods, at a reduced resolution, to compare the distribution of errors across FCE and the synthetic corpora. These are presented in Table 3.

At a high level, NMT generates errors more often among more common parts-of-speech, favouring errors in verbs and nouns, rather than in adverbs and conjunctions. It did not make spelling errors as often as in the source dataset; this is likely because it only observed the specific spelling errors present in FCE, and as the vocabulary is restricted to that dataset, it does not encounter those words as frequently in EVP, and thus rarely makes the same spelling mistakes.

Additionally, the differences in these distributions can partially be attributed to the implicit differences between us and the annotators of FCE.

Spelling	FCE	AM	TS	BS
Spelling errors	11	1	1	4

Part-of-speech	FCE	AM	TS	BS
Verb	34	16	26	16
Preposition	18	16	10	14
Determiner	16	7	6	10
Noun	13	36	35	43
Pronoun	7	3	3	1
Adverb	5	5	3	5
Adjective	3	15	16	12
Conjunction	2	2	2	1
Quantifier	1	0	0	0

Remedy Type	FCE	AM	TS	BS
Replacement	49	35	34	32
Inclusion	23	30	27	35
Removal	14	33	36	32
Word form	9	2	2	1
Word order	5	0	0	0

Table 3: Error distribution across FCE and manually annotated samples of artificial data. Spelling errors are a % of all errors, while Part-of-speech, and Remedy Type are compared within their own categories to sum to 100%.

5.2 Comparison with human errors

To check if the synthetic instances passed for human-like, we mixed 50 generated sentences among an equal number of actual ungrammatical instances from FCE-dev and tasked a human evaluator to identify the artificial statements, in a simple Turing-style test. We created three such sets, one for each of our sampling techniques, and the test subject aimed to identify synthetic samples with high confidence. Results of this test are presented in Table 4.

The high precision but low recall scores suggest that while it is still possible to spot some corruptions that are quite clearly artificial, the bulk of our samples do not betray their synthetic nature and are indistinguishable from naturally occurring erroneous sentences. In order to fairly compare our work with earlier results, we intended to conduct such a test for sentences generated by the SMT of Rei et al. (2017). Unfortunately, we were only able to source corruptions of FCE-train via this method; therefore, we decided not to perform this test as its results cannot be compared to ours.

	AM	TS	BS
Precision	81.25	63.63	50.00
Recall	26.00	28.00	14.00
F1	39.39	38.89	22.22

Table 4: Results of a Turing-style test, where a subject was asked to distinguish between real and fake sentences, sampled from each of the different generated corpora.

6 Conclusions and future work

We presented a novel data augmentation technique for grammatical error detection using neural machine translation to learn the distribution of language-learner errors, and induce such errors into grammatically correct text. We explored several different variants of sampling to improve the quality of our synthetic errors. After creating artificial training instances with an off-the-shelf NMT, we bettered previous state-of-the-art results on the canonical test with even a basic BiLSTM, and established a new state of the art using a stronger model. Additionally, we demonstrated that we were able to leverage corruptions of an out-of-domain dataset to set new benchmarks on separate, also out-of-domain tests, without specifically optimising for either.

Our work indicates that neural error generation warrants further investigation with different datasets and architectures, both for error detection and error correction. Among possible future work is using generative adversarial networks as corruption engines, and developing better sequence alignment methods. Some preliminary results with simple corruptions using word substitution and word dropout (Iyyer et al., 2015) appear to be promising, and may feature as components of a future corruption system. Finally, one could use such artificial error-prone corpora as source text for self-training an error detection system.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Guillaume Bouchard, Pontus Stenetorp, and Sebastian Riedel. 2016. Learning to generate textual data. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1608–1616, Austin, Texas. Association for Computational Linguistics.

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive Exploration of Neural Machine Translation Architectures. *ArXiv e-prints*.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. Association for Computational Linguistics.

Jennifer Foster and Øistein E Andersen. 2009. Generrate: generating errors for use in grammatical error detection. In *Proceedings of the fourth workshop on innovative use of nlp for building educational applications*, pages 82–90. Association for Computational Linguistics.

Andrew R Golding and Dan Roth. 1996. Applying winnow to context-sensitive spelling correction. *arXiv preprint cmp-lg/9607024*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

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

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 388–392. Association for Computational Linguistics.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. The amu system in the conll-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, Baltimore, Maryland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. Association for Computational Linguistics.

Kevin Knight, Ishwar Chander, Matthew Haines, Vasileios Hatzivassiloglou, Eduard Hovy, Masayo Iida, Steve K Luk, Akitoshi Okumura, Richard Whitney, and Kenji Yamada. 1994. Integrating knowledge bases and statistics in mt. *arXiv preprint cmp-lg/9409001*.

Nina H Macdonald. 1983. Human factors and behavioral science: The unix writer's workbench software: Rationale and design. *Bell Labs Technical Journal*, 62(6):1891–1908.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

Courtney Napoles and Chris Callison-Burch. 2017. Systematically adapting machine translation for grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 345–356, Copenhagen, Denmark. Association for Computational Linguistics.

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*, volume 16, pages 572–581.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the*

55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, Osaka, Japan. The COLING 2016 Organizing Committee.

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*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Stephen D. Richardson and Lisa C. Braden-Harder. 1988. The experience of developing a large-scale natural language text processing system: Critique. In *Second Conference on Applied Natural Language Processing*.

Alla Rozovskaya and Dan Roth. 2010. Training paradigms for correcting errors in grammar and usage. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 154–162. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

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 (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Jason Wang and Luis Perez. 2017. The effectiveness of data augmentation in image classification using deep learning. Technical report, Technical report.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806, Copenhagen, Denmark. Association for Computational Linguistics.

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

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria. Association for Computational Linguistics.