The Unreasonable Effectiveness of Transformer Language Models in Grammatical Error Correction

Dimitris Alikaniotis   **Vipul Raheja**

Grammarly

New York City, NY

`firstname.lastname@grammarly.com`

Abstract

Recent work on Grammatical Error Correction (GEC) has highlighted the importance of language modeling in that it is certainly possible to achieve good performance by comparing the probabilities of the proposed edits. At the same time, advancements in language modeling have managed to generate linguistic output, which is almost indistinguishable from that of human-generated text. In this paper, we up the ante by exploring the potential of more sophisticated language models in GEC and offer some key insights on their strengths and weaknesses. We show that, in line with recent results in other NLP tasks, Transformer architectures achieve consistently high performance and provide a competitive baseline for future machine learning models.

1 Introduction

Transformer models (Vaswani et al., 2017) trained on large-scale language modeling datasets have recently proved to be a very effective means of representing the meaning of a sentence, being put to effective use in fine-tuning both sentence-level tasks, such as the GLUE benchmark (Wang et al., 2018) and token-level tasks, such as Named Entity Recognition (Devlin et al., 2019). Recent work has also found them to produce linguistically valid representations (Goldberg, 2019), as well as to display excellent performance across multiple downstream NLP tasks (e.g., Houlsby et al. 2019).

In this work, we explore how such models perform in the task of Grammatical Error Correction (GEC). While there is a substantial amount of work on statistical (Rozovskaya and Roth, 2016; Junczys-Dowmunt and Grundkiewicz, 2014; Yannakoudakis et al., 2017) and neural (Ji et al., 2017; Xie et al., 2016; Yuan and Briscoe, 2016; Chollampatt et al., 2016; Chollampatt and Ng, 2017; Sakaguchi et al., 2017; Chollampatt and

Ng, 2018) machine translation methods for GEC, we follow the approach of Bryant and Briscoe (2018) and explore how such models would fare in this task when treated as simple language models. More specifically, Bryant and Briscoe (2018) train a 5-gram language model on the One Billion Word Benchmark (Chelba et al., 2013) dataset and find that it produces competitive baseline results without any supervised training. In our work, we extend this work by substituting the *n*-gram model for several publicly available implementations of state-of-the-art Transformer language models trained on large linguistic corpora and assess their performance on GEC without any supervised training. We find that Transformer language models produce results on par with supervised approaches providing a solid baseline system. This finding is of particular importance in GEC, where data collection and annotation requires substantial manual effort.

2 Related Work

The idea of using language models is quite fundamental to the task of Grammatical Error Correction, which has fed a substantial body of work over the years. More recently, with the availability of web-scale data powering the advances in language modeling, among most of the other advances in NLP, a plethora of language-modeling based approaches have been proposed for the GEC task. Gamon et al. (2008); Matthieu Hermet and Szpakowicz (2008) and Yi et al. (2008) were some of the early works to successfully leverage language models trained on large amounts of web-scale data into a GEC system, reinforcing the idea that simple models and a lot of data trump more elaborate models based on annotated data (Halevy et al., 2009).

Since then, multiple works based on language-

models have been proposed for the GEC task (Park and Levy, 2011; Dahlmeier and Ng, 2012a), either relying entirely on LMs or using them for fine-tuning their systems. Many of the top-ranked systems in the CoNLL-2013 and 2014 GEC shared tasks (Ng et al., 2013, 2014), were either based on language models or had them as integral parts of their systems (Kao et al., 2013; Yoshimoto et al., 2013; Xing et al., 2013; Lee and Lee, 2014; Junczys-Dowmunt and Grundkiewicz, 2014). LM-only approaches though took a backseat and were only sporadically used after the shared tasks, as Neural Machine Translation-based approaches took over, but LMs remained an integral part of the GEC systems (Junczys-Dowmunt and Grundkiewicz, 2016; Ji et al., 2017; Xie et al., 2016; Junczys-Dowmunt et al., 2018; Chollampatt and Ng, 2018). However, Bryant and Briscoe (2018) recently revived the idea, achieving competitive performance with the state-of-the-art, demonstrating the effectiveness of the approaches to the task without using any annotated data for training.

3 Methodology

In this work, we follow the setup from Bryant and Briscoe (2018) substituting the 5-gram language model for different language models based on the Transformer architecture. Specifically, we use Google's BERT (Devlin et al., 2019) and OpenAI's GPT (Radford et al., 2018) and GPT-2 (Radford et al., 2019).[1] While all these are best thought of as language models in that they have been trained to predict an element in a sequence, they use slightly different objectives which does not make them directly comparable. Specifically, GPT and GPT-2 have been trained with a classic language modeling objective, whereby they predict the next word in a sequence, whereas BERT has been trained using a *masked language modeling* objective in which the network attempts to predict masked words in the sentence.

We extract the probability of a sentence from BERT, by iteratively masking every word in the sentence and then summing the log probabilities. While this approach is far from ideal, it has been shown (Wang and Cho, 2019) that it approximates the log-likelihood of a sentence.

Test set	Sent.	Tokens	Annot.
CoNLL-2014	1,312	30k	2
FCE	2,715	47k	1

Table 1: Statistics for evaluation data

3.1 Confusion sets

Since our systems do not generate novel sequences, we follow Bryant and Briscoe (2018) and use simple heuristics to generate a confusion set of sentences that our language models score. For prepositions and determiners, the confusion set includes the set of all prepositions and determiners plus an empty string ϵ to remove unnecessary additions. For morphological errors (e.g., past tense or pluralization), we use the Automatically Generated Inflection Database (AGID) which contains different morphological forms for each word.[2] However, we notice that due to the automatic generation, AGID contains errors that might propagate into our scoring. The problem with introducing new errors and non-words is that they would be interpreted as unknown words (henceforth [UNK]s) from the model's perspective. An unknown word in some context might give higher probabilities to an erroneous sentence and cause the model not to select the correct alternative. To remedy this issue, we generate a vocabulary from all the training sets and make sure that any proposed words which do not exist in the vocabulary are replaced by [UNK]s. Note that there is no reason to re-use the vocabulary of the training sets as any large English wordlist would achieve a similar effect. Finally, for spelling mistakes, we, again, follow Bryant and Briscoe (2018) and use CyHunSpell[3] to generate alternatives for non-words.

3.2 Thresholding

Given that our confusion set is prone to errors (due to its automatic generation procedure) as well as the fact that we cannot target all potential errors (e.g., insertions), we bias our method to *prefer* the original sentence unless a much better the alternative is found. We quantify this margin by imposing a threshold above which we accept a candidate sentence as a better alternative. Concretely, let $P(s_c)$ be the probability of the candidate sentence and $P(s_o)$ the probability of the

---

[1]https://github.com/huggingface/pytorch-pretrained-BERT/

[2]http://wordlist.aspell.net/other/
[3]https://pypi.org/project/CyHunspell/

		ERRANT			M2		
Dataset	System	P	R	$F_{0.5}$	P	R	$F_{0.5}$
CoNLL-2014	Felice et al. (2014) †	-	-	-	39.71	30.10	37.33
	Yannakoudakis et al. (2017)	-	-	-	58.79	30.63	49.66
	Chollampatt and Ng (2017)	-	-	-	62.74	32.96	53.14
	Chollampatt and Ng (2018)	-	-	-	65.49	33.14	54.79
	Ge et al. (2018)	-	-	-	**74.12**	**36.30**	**61.34**
	Bryant and Briscoe (2018)	36.62	19.93	31.37	40.56	20.81	34.09
	BERT	33.27	**27.14**	31.83	35.69	**27.99**	33.83
	GPT-1	49.58	27.06	42.5	51.08	27.45	43.57
	GPT-2	**57.73**	24.75	**45.58**	**58.51**	24.9	**46.08**
FCE	Yannakoudakis et al. (2017)	-	-	-	**65.03**	32.45	**54.15**
	Bryant and Briscoe (2018)	41.92	13.62	29.61	44.78	14.12	31.22
	BERT	29.56	**34.67**	30.46	31.97	**35.01**	32.53
	GPT-1	**62.75**	32.19	52.74	**64.01**	32.33	**53.52**
	GPT-2	61.91	33.47	**52.92**	62.64	33.74	53.48

Table 2: Results of our Transformer-Language Model approach against similar approaches (Bryant and Briscoe, 2018) and state-of-the-art on Grammatical Error Correction. For each of the datasets, we use the corresponding test set, and we do not train our models on the corpora. As BERT, we report the best performing BERT model (12 layers, retaining uppercase characters). In the top part of each dataset, we report the scores of supervised methods and in the bottom the unsupervised ones. † denotes this system won the shared task competition.

original sentence, then we accept the candidate if $P(s_c) > P(s_o) + \tau$, where τ is some threshold parameter which we fit on each development set. Note that, practically, this parameter controls the trade-off between *precision* and *recall* as higher τ values would mean that there is less chance of changing the original sentence (i.e., higher precision) and vice versa. We explore different values for $\tau \in \{0, 2, 4, 6, 8\}$ by, as above, fitting them on the corresponding development set.[4]

3.3 Search

Finally, we perform *greedy search* to find the best alternative sentence by iterating over each sentence multiple times, once for every position for which our heuristics found alternatives. If an alternative is selected for the target position, we update the original sentence and proceed to the next position. This pseudo-log-likelihood approximation makes the problem of considering every permutation more computationally tractable.

4 Experiments

We evaluate our method and report results on two standard publicly available datasets. Our evaluation is aimed to stay as true to Bryant and Briscoe (2018) as possible to ensure an even comparison. Concretely, we use the test dataset from

the CoNLL-2014 (Ng et al., 2014) shared task[5] and the publicly available First Certificate in English (FCE) (Yannakoudakis et al., 2011). Unfortunately, due to licensing issues, we were unable to obtain permission to use the JFLEG (Napoles et al., 2017) corpus for evaluation. Note that in our method, we do not make use of the training sets commonly used with these datasets. However, we use the development sets used by Bryant and Briscoe (2018) to tune the hyperparameter τ. The number of sentences and tokens for the datasets we used can be found in Table 1.

Similar to Bryant and Briscoe (2018), we report results on three metrics. We use the MaxMatch (M^2) Precision, Recall and $F_{0.5}$ (Dahlmeier and Ng, 2012b) and ERRANT Precision, Recall and $F_{0.5}$ (Bryant et al., 2017).

5 Results

Table 2 presents the results of our method comparing them against recent state-of-the-art supervised models and the simple *n*-gram language model used by Bryant and Briscoe (2018). Table 3 shows some qualitative examples on how each model corrects two sentences pulled from the FCE along with the gold annotations. The reported results

---

[4]Note that the probability of each sentence is in log space.

[5]While we acknowledge the contemporaneous nature of the BEA 2019 Shared Task on GEC and would have liked to report results on the W&I+LOCNESS data, we could not do so because of license limitations.

Source	It will start by a speech from the Director of the conference, followed by a meal.
Gold	It will start **with** a speech **by** the Director of the conference, followed by a meal.
BERT	It will start **with** a speech **from** the Director of the conference, followed by a meal.
GPT	It will start **by** a speech **from** the Director of the conference, followed by a meal.
GPT-2	It will start **with** a speech **from** the Director of the conference, followed by a meal.
Source	They all knows where the conference is and when.
Gold	They all **know** where the conference is and when.
BERT	They all know where the **conferencing** is and when.
GPT	They all **knows** where the conference is and when.
GPT-2	They all **know** where the conference is and when.

Table 3: Source sentences along with the gold edits and the proposed candidates from each of our models.

come from the best performing hyperparameter τ on each dataset. For BERT, we also explored different sizes (12 vs. 24 layers) and whether retaining uppercase characters helps in performance. The best performing τ values were $\tau = 4$ for CoNLL14 for all models; for the FCE dataset: BERT $\tau = 4$, GPT $\tau = 8$, and GPT-2 $\tau = 6$. The best 'version,' of BERT was the *large*, *cased* (i.e., retaining the lower- /uppercase distinction).

A key result of Table 2 is that Transformer Language Models prove to be more than just a competitive baseline to legitimate Grammatical Error Correction systems on their own. Across the board, Transformer Models are able to outperform the simple *n*-gram model and even approach the performance of supervised GEC systems.

6 Discussion

Looking at the performance of the two GPT models more closely, we see that their performance is nearly identical with GPT-2 leading by a small margin in the CoNLL14 dataset. Given that the versions we used share the same number of layers (12), we attribute GPT-2's slight advantage to the fact that it was trained on considerably more data.

Another interesting result is that while BERT surpasses the *n*-gram baseline overall, it achieves worse performance than the rest in terms of precision and $F_{0.5}$ score. Considering its overall success at modeling NLP tasks, one might expect BERT to achieve better performance here. However, as mentioned above, BERT is not truly a language model in the sense that GPT and GPT-2 are but uses a quasi-language modeling objective which could explain its degraded performance in this setting. Note that framing the task differently (e.g., by masking the preposition in a sentence and selecting the one with the highest probability) might give the edge to BERT as it resembles the way it was trained.

It is also worth mentioning that despite tuning τ to each dataset, we do not explore different weights for different kinds of errors (e.g., penalizing more spelling mistakes). Our key motivation was to corroborate and extend the results of Bryant and Briscoe (2018) to current state-of-the-art language models which have been trained in several languages and show that these models are tough baselines to beat for novel GEC systems.

While the results of the Transformer language models shown in Table 2 demonstrate that they are a tough baseline to beat, it is worth noting that the present approach is not without its limitations. We believe that our methodology should not be considered a panacea to GEC. For instance, being bound by the confusion sets, our system (1) cannot handle missing words (which make up about 20% of all errors), and (2) it is tuned to capture only a subset of the possible mistakes a writer can make (closed class words).

It could be argued that since our system makes use of a pre-defined confusion set (even an automatically generated one), it could not be considered as a fully unsupervised system. In principle, we agree with that statement and we believe that a system which uses, for example, corpus statistics to on-the-fly generate a confusion set would be a very interesting exercise and could yield similar results. However, the present paper is concerned with highlighting the importance of language modeling in GEC and its potential in aiding in low-resource languages where large parallel datasets are unavailable, but such confusion sets are relatively easily obtainable.

7 Conclusion

In this work, we advanced on the foundational idea that a simple language modeling-based approach to GEC with no annotated data can challenge the latest neural and machine translation approaches that rely on large quantities of annotated training data. To this end, we improve on previous work by leveraging state-of-the-art language modeling techniques and perform a thorough comparison of three state-of-the-art Transformer language models which in turn have been trained on data of the order of hundreds of millions of words. We find that merely using pre-trained, and publicly available neural language models improves the performance by a significant margin and comes within striking distance of the state-of-the-art methods.

This work reinforces the strength and robustness of language-model based methods for the task of grammatical error correction. While recent state-of-the-art GEC systems are pursuing NMT-based models with huge amounts (millions of sentences) of annotated training data, approaches like this which require no annotated training data provide great value to researchers and developers interested in building competitive GEC systems (e.g., in other languages) with limited annotated data.

Acknowledgements

References

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253, New Orleans, Louisiana. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google.

Shamil Chollampatt and Hwee Tou Ng. 2017. Connecting the dots: Towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 327–333, Copenhagen, Denmark. Association for Computational Linguistics.

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*.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, USA.

Daniel Dahlmeier and Hwee Tou Ng. 2012a. A beam-search decoder for grammatical error correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578, Jeju Island, Korea. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

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

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

A. Halevy, P. Norvig, and F. Pereira. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.

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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Ting-hui Kao, Yu-wei Chang, Hsun-wen Chiu, Tzu-Hsi Yen, Joanne Boisson, Jian-cheng Wu, and Jason S. Chang. 2013. CoNLL-2013 shared task: Grammatical error correction NTHU system description. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 20–25, Sofia, Bulgaria. Association for Computational Linguistics.

Kyusong Lee and Gary Geunbae Lee. 2014. POSTECH grammatical error correction system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 65–73, Baltimore, Maryland. Association for Computational Linguistics.

Alain Dsilets Matthieu Hermet and Stan Szpakowicz. 2008. Using the web as a linguistic resource to automatically correct lexico-syntactic errors. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. Jfleg: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. 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.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Y. Albert Park and Roger Levy. 2011. Automated whole sentence grammar correction using a noisy channel model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 934–944, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2205–2215, Berlin, Germany. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 366–372, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a markov random field language model. *CoRR*, abs/1902.04094.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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

Junwen Xing, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Xiaodong Zeng. 2013. UM-checker: A hybrid system for English grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 34–42, Sofia, Bulgaria. Association for Computational Linguistics.

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.

Xing Yi, Jianfeng Gao, and William B. Dolan. 2008. A web-based English proofing system for English as a second language users. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2013. NAIST at 2013 CoNLL grammatical error correction shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 26–33, Sofia, Bulgaria. 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.