Language Model Based Grammatical Error Correction without Annotated Training Data

Christopher Bryant **Ted Briscoe**
ALTA Institute
Department of Computer Science and Technology
University of Cambridge
Cambridge, UK
{cjb255, ejb1}@cl.cam.ac.uk

Abstract

Since the end of the CoNLL-2014 shared task on grammatical error correction (GEC), research into language model (LM) based approaches to GEC has largely stagnated. In this paper, we re-examine LMs in GEC and show that it is entirely possible to build a simple system that not only requires minimal annotated data (∼1000 sentences), but is also fairly competitive with several state-of-the-art systems. This approach should be of particular interest for languages where very little annotated training data exists, although we also hope to use it as a baseline to motivate future research.

1 Introduction

In the CoNLL-2014 shared task on Grammatical Error Correction (GEC) (Ng et al., 2014), the top three teams all employed a combination of statistical machine translation (SMT) or classifier-based approaches (Junczys-Dowmunt and Grundkiewicz, 2014; Felice et al., 2014; Rozovskaya et al., 2014). These approaches have since come to dominate the field, and a lot of recent research has focused on fine-tuning SMT systems (Junczys-Dowmunt and Grundkiewicz, 2016), reranking SMT output (Hoang et al., 2016; Yuan et al., 2016), combining SMT and classifier systems (Susanto et al., 2014; Rozovskaya and Roth, 2016), and developing various neural architectures (Chollampatt et al., 2016; Xie et al., 2016; Yuan and Briscoe, 2016; Chollampatt and Ng, 2017; Sakaguchi et al., 2017; Yannakoudakis et al., 2017).

Despite coming a fairly competitive fourth in the shared task however (Lee and Lee, 2014), research into language model (LM) based approaches to GEC has largely stagnated. The main aim of this paper is hence to re-examine language modelling in the context of GEC and show that it is still possible to achieve competitive results even with very simple systems. In fact, a notable strength of LM-based approaches is that they rely on very little annotated data (purely for tuning purposes), and so it is entirely possible to build a reasonable correction system for any language given enough native text. In contrast, this is simply not possible with SMT and other popular approaches which always require (lots of) labelled data.

2 Methodology

The core idea behind language modelling in GEC is that low probability sequences are more likely to contain grammatical errors than high probability sequences. For example, **discuss about the problem* is expected to be a low probability sequence because it contains an error while *discuss the problem* or *talk about the problem* are expected to be higher probability sequences because they do not contain errors. The goal of LM-based GEC is hence to determine how to transform the former into the latter based on LM probabilities.[1]

With this in mind, our approach is fundamentally a simplification of the algorithm proposed by Dahlmeier and Ng (2012a). It consists of 5 steps and is illustrated in Table 1:

1. Calculate the normalised log probability of an input sentence.
2. Build a confusion set, if any, for each token in that sentence.
3. Re-score the sentence substituting each candidate in each confusion set.
4. Apply the single best correction that increases the probability above a threshold.
5. Iterate steps 1-4.

One of the main contributions of this paper is hence to re-evaluate the LM approach in relation to the latest state-of-the-art systems on several benchmark datasets.

[1] See Chelba et al. (2014) for more information about popular approaches to language modelling.

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 247–253
New Orleans, Louisiana, June 5, 2018.

Step	Sentence																Probability
1	I		am		looking		forway		to		see		you		soon		.		-2.71
2 and 3	I	was	-2.67	look	-2.91	**forward**	**-1.80**	of	-2.98	seeing	-3.09	you	sooner	-3.05	.		-
		be	-3.09	looks	-2.93	Norway	-2.36	in	-2.99	saw	-3.25		soonest	-3.20			
		are	-3.10	looked	-2.95	foray	-2.70	ϵ	-3.00	sees	-3.39						
		...		...		...		...		...							
4	I		am		looking		**forward**		to		see		you		soon		.		-1.80
5	I		am		looking		forward		to		**seeing**		you		soon		.		-1.65

Table 1: A step-by-step example of our approach as described in Section 2. All scores are log probabilities.

2.1 Sequence Probabilities

We evaluate hypothesis corrections in terms of normalised log probabilities at the sentence level. Normalisation by sentence length is necessary to overcome the tendency for shorter sequences to have higher probabilities than longer sequences. Dahlmeier and Ng (2012a) similarly used normalised log probabilities to evaluate hypotheses, but did so as part of a more complex combination of other features. In contrast, Lee and Lee (2014) evaluated hypotheses in terms of sliding five word windows (5-grams).

2.2 Confusion Sets

One of the defining characteristics of LM-based GEC is that the approach does not necessarily require annotated training data. For example, spellcheckers and rules both formed key parts of Dahlmeier and Ng's and Lee and Lee's systems. While Lee and Lee ultimately did make use of annotated training data however, Dahlmeier and Ng instead employed separate classifiers for articles, prepositions and noun number errors trained only on native text.

In this work, we focus on correcting the following error types in English: non-words, morphology, and articles and prepositions.[2]

Non-words: We use CyHunspell[3] v1.2.1 with the latest British English Hunspell dictionaries[4] to generate correction candidates for non-word errors. Non-words include genuine misspellings, such as [*freind* → *friend*], and inflectional errors, such as [*advices* → *advice*]. Although CyHunspell is not a context sensitive spell checker, the proposed corrections are evaluated in a context sensitive manner by the language model.

Morphology: Examples of morphological errors include noun number [*cat* → *cats*], verb tense [*eat* → *ate*] and adjective form [*big* → *bigger*], amongst others. To generate correction candidates for morphological errors, we use an Automatically Generated Inflection Database (AGID),[5] which contains all the morphological forms of many English words. The confusion set for a word is hence derived from this database.

Articles and Prepositions: Since articles and prepositions are closed class words, we defined confusion sets for these error types manually. Specifically, the article confusion set consists of $\{\epsilon$, a, an, the$\}$, while the preposition confusion set consists of the top ten most frequent prepositions: $\{\epsilon$, about, at, by, for, from, in, of, on, to, with$\}$. Both sets also contain a null character which represents a deletion.

Unlike Dahlmeier and Ng and Lee and Lee, we do not yet handle missing words ($\sim$20% of all errors) because it is often difficult to know where to insert them.

2.3 Iteration

The main reason to iteratively correct only one word at a time is because errors sometimes interact. For example, correcting [*see* → *seeing*] in Table 1 initially reduces the log probability of the input sentence from -2.71 to -3.09. After correcting [*foray* → *forward*] however, [*see* → *seeing*] subsequently increases the probability of the sentence from -1.80 to -1.65 in the second iteration. Consequently, correcting the most serious errors first, in terms of language model probability increase, often helps facilitate the correction of less serious errors later. Dahlmeier and Ng and Lee and Lee both also used iterative correction strategies in their systems, but did so as part of a beam search or pipeline approach respectively.

[2] Note that targeting other error types may be more appropriate in other languages; e.g. Mandarin Chinese contains very little morphology.

[3] https://pypi.python.org/pypi/CyHunspell

[4] https://sourceforge.net/projects/wordlist/files/speller/2017.08.24/

[5] http://wordlist.aspell.net/other/

Dataset	Tokenizer	Sents	Coders	Edits
CoNLL-2013	NLTK	1381	1	3404
CoNLL-2014	NLTK	1312	2	6104
FCE-dev	spaCy	2371	1	4419
FCE-test	spaCy	2805	1	5556
JFLEG-dev	NLTK	754	4	10576
JFLEG-test	NLTK	747	4	10082

Table 2: Various stats about the learner corpora we use.

3 Data and Resources

In all our experiments, we used a 5-gram language model trained on the One Billion Word Benchmark dataset (Chelba et al., 2014) with KenLM (Heafield, 2011). While a neural model would likely result in better performance, efficient training on such a large amount of data is still an active area of research (Grave et al., 2017).

Although LM-based GEC does not require annotated training data, a small amount of annotated data is still required for development and testing. We hence make use of several popular GEC corpora, including: CoNLL-2013 and CoNLL-2014 (Ng et al., 2013, 2014), the public First Certificate in English (FCE) (Yannakoudakis et al., 2011), and JFLEG (Napoles et al., 2017).

Since the FCE was not originally released with an official development set, we use the same split as Rei and Yannakoudakis (2016),[6] which we tokenize with spaCy[7] v1.9.0. We also reprocess all the datasets with the ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017) in an effort to standardise them. This standardisation is especially important for JFLEG which is not explicitly annotated and so otherwise cannot be evaluated in terms of F-score. Note that results on CoNLL-2014 and JFLEG are typically higher than on other datasets because they contain more than one reference. See Table 2 for more information about each of the development and test sets.

4 Tuning

The goal of tuning in our LM-based approach is to determine a probability threshold that optimises $F_{0.5}$. For example, although the edit [*am* → *was*] in Table 1 increases the normalised sentence log probability from -2.71 to -2.67, this is such a small improvement that it is likely to be a false positive. In order to minimise false positives, we hence set a threshold such that a candidate correction must improve the average token probability of the sentence by at least X% before it is applied. Although it may be unusual to use percentages in log space, this is just one way to compare the difference between two sentences which we found worked well in practice.

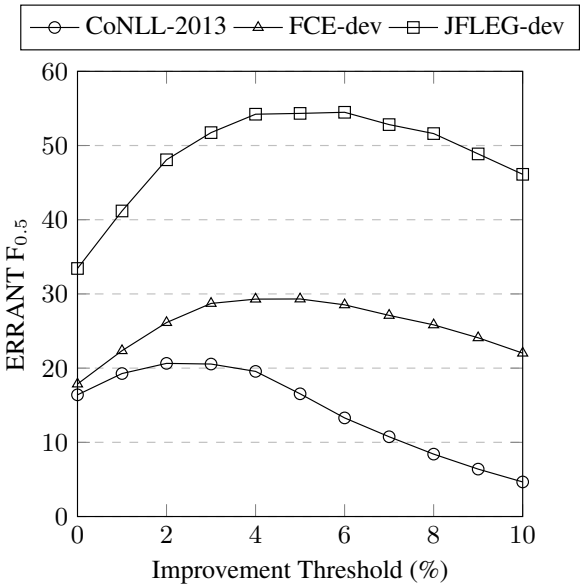

Figure 1: The effect of changing the sentence probability improvement threshold (%) on ERRANT $F_{0.5}$ for each of the development sets.

The results of this tuning are shown in Figure 1, where we tried thresholds in the range of 0-10% on three different development sets. It is notable that the optimum threshold for CoNLL-2013 (2%) is very different from that of FCE-dev (4%) and JFLEG-dev (5%), which we suspect is because each dataset has a different error type distribution. For example, spelling errors make up just 0.3% of all errors in CoNLL-2013, but closer to 10% in FCE-dev and JFLEG-dev.

Finally, it should be noted that this threshold is an approximation and it is certainly possible to optimise further. For example, in future, thresholds could be set based on error types rather than globally.

5 Results and Discussion

Before evaluating performance on the test sets, a final post-processing step changed the first alphabetical character of every sentence to upper case if necessary. This improved the scores by about 0.3 $F_{0.5}$ in CoNLL-2014 and FCE-test, but by over 5 $F_{0.5}$ in JFLEG-test. This surprising result once

[6] https://ilexir.co.uk/datasets/index.html

[7] https://spacy.io/

		ERRANT			M2 Scorer			
Test Set	**System**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**GLEU**
CoNLL-2014	Lee and Lee (2014)	30.60	20.95	28.02	34.51	21.73	30.88	59.50
	$AMU16_{SMT}$+LSTM	-	-	-	**58.79**	**30.63**	**49.66**	**68.26**
	$CAMB16_{SMT}$+LSTM	-	-	-	49.58	21.84	39.53	65.68
	Our work	36.62	19.93	31.37	40.56	20.81	34.09	59.35
FCE-test	$AMU16_{SMT}$ + LSTM	-	-	-	40.67	17.36	32.06	63.57
	$CAMB16_{SMT}$ + LSTM	-	-	-	**65.03**	**32.45**	**54.15**	**70.72**
	Our work	41.92	13.62	29.61	44.78	14.12	31.22	60.04
JFLEG-test	$AMU16_{SMT}$ + LSTM	-	-	-	60.68	22.65	45.43	42.65
	$CAMB16_{SMT}$ + LSTM	-	-	-	65.86	30.56	53.50	46.74
	Sakaguchi et al. (2017)	-	-	-	65.80	**40.96**	**58.68**	**53.98**
	Our work	73.76	27.61	55.28	**76.23**	28.48	57.08	48.75

Table 3: Our LM-based approach is compared against several state-of-the-art results. $AMU16_{SMT}$+LSTM and $CAMB16_{SMT}$+LSTM were both originally reported by Yannakoudakis et al. (2017), while Lee and Lee (2014) is the system entered by POST in CoNLL-2014. Only our approach does **not** use annotated training data.

again shows that different test sets have very different error type distributions and that even the simplest of correction strategies can significantly affect results.

Our final scores are shown in Table 3 where they are compared with several state-of-the-art systems. Unfortunately, we cannot compare results with Dahlmeier and Ng (2012a) because this system is neither publicly available nor has previously been evaluated on these test sets. Results are reported in terms of M2 $F_{0.5}$ (Dahlmeier and Ng, 2012b), the *de facto* standard of GEC evaluation; ERRANT $F_{0.5}$ (Bryant et al., 2017), an improved version of M2 which we used to develop our system; and GLEU (Napoles et al., 2015), an ngram-based metric designed to correlate with human judgements. Results for ERRANT are not available in all cases because system output is not available.

At this point, it is worth reiterating that our main intention was not to necessarily improve upon the state-of-the-art, but rather quantify the extent to which a simple LM-based approach with minimal annotated data could compete against a much more sophisticated model trained on millions of words of annotated text. This is especially relevant for languages where annotated training data may not be available.

With this in mind, we were firstly pleased to improve upon the previous best LM-based approach by Lee and Lee (2014) in the CoNLL-2014 shared task. This is especially significant given we also did so without any annotated training data (unlike them). Although our system would still have placed fourth overall, the gap between third and fourth decreased from 3 $F_{0.5}$ to less than 1 $F_{0.5}$.

We were also surprised by the high performance on JFLEG-test, where we not only outperformed two state-of-the-art systems, but also came to within 2 $F_{0.5}$ of the top system. This is especially surprising given our system only corrects a limited number of error types (roughly 14 out of the 55 in ERRANT[8]), and so can maximally correct only 40-60% of all errors in each test set. One possible explanation for this is that unlike CoNLL-2014 and FCE-test, which were only corrected with minimal edits, JFLEG was corrected for fluency (Sakaguchi et al., 2016), and so it intuitively makes sense that LM-based approaches perform better with fluent references.

Although we did not perform as well on CoNLL-2014 or FCE-test, most likely for the same reason, we also note a large discrepancy between state-of-the-art systems tuned on different datasets. For example, while $AMU16_{SMT}$+LSTM tuned for CoNLL achieves the highest result on CoNLL-2014 (49.66 $F_{0.5}$), its equivalent performance on FCE-test (32.06 $F_{0.5}$) is only marginally better than our own (31.22 $F_{0.5}$). We observe a similar effect with $CAMB16_{SMT}$+LSTM tuned for the FCE, and so are wary of approaches that might be overfitting to their training corpora.

We make all our code and system output available online.[9]

[8] R:ADJ:FORM, R:DET, R:MORPH, R:NOUN:INFL, R:NOUN:NUM, R:ORTH, R:PREP, R:SPELL, R:VERB:FORM, R:VERB:INFL, R:VERB:SVA, R:VERB:TENSE, U:DET, U:PREP

[9] https://github.com/chrisjbryant/lmgec-lite

6 Conclusion

In this paper, we have shown that a simple language model approach to grammatical error correction with minimal annotated data can still be competitive with the latest neural and machine translation approaches that rely on large quantities of annotated training data. This is especially significant given that our system is also limited by the range of error types it can correct. In the future, we hope to improve our system by adding the capability to correct other error types, such as missing words, and also make use of neural language modelling techniques.

We have demonstrated that LM-based GEC is not only still a promising area of research, but one that may be of particular interest to researchers working on languages where annotated training corpora are not yet available. We released all our code and system output with this paper.

References

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)*. Association for Computational Linguistics, Vancouver, Canada, pages 793–805. http://aclweb.org/anthology/P17-1074.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. pages 2635–2639. http://www.isca-speech.org/archive/interspeech_2014/i14_2635.html.

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016. Adapting grammatical error correction based on the native language of writers with neural network joint models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1901–1911. https://aclweb.org/anthology/D16-1195.

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*. Association for Computational Linguistics, Copenhagen, Denmark, pages 327–333. http://www.aclweb.org/anthology/W17-5037.

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*. Association for Computational Linguistics, Jeju Island, Korea, pages 568–578. http://www.aclweb.org/anthology/D12-1052.

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*. Association for Computational Linguistics, Montréal, Canada, pages 568–572. http://www.aclweb.org/anthology/N12-1067.

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*. Association for Computational Linguistics, Baltimore, Maryland, pages 15–24. http://www.aclweb.org/anthology/W14-1702.

Édouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017. Efficient softmax approximation for GPUs. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*. PMLR, International Convention Centre, Sydney, Australia, volume 70 of *Proceedings of Machine Learning Research*, pages 1302–1310. http://proceedings.mlr.press/v70/grave17a.html.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, United Kingdom, pages 187–197. https://kheafield.com/papers/avenue/kenlm.pdf.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an smt approach to grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. AAAI Press / International Joint Conferences on Artificial Intelligence, New York, New York, USA, pages 2803–2809. https://www.ijcai.org/Proceedings/16/Papers/398.pdf.

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. Association for Computational Linguistics, Baltimore, Maryland, pages 25–33. http://www.aclweb.org/anthology/W14-1703.

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*. Association for Computational Linguistics, Austin, Texas, pages 1546–1556. https://aclweb.org/anthology/D16-1161.

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*. Association for Computational Linguistics, Baltimore, Maryland, pages 65–73. http://www.aclweb.org/anthology/W14-1709.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 588–593. http://www.aclweb.org/anthology/P15-2097.

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*. Association for Computational Linguistics, Valencia, Spain, pages 229–234. http://www.aclweb.org/anthology/E17-2037.

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*. ACL, Baltimore, Maryland, USA, pages 1–14. http://aclweb.org/anthology/W/W14/W14-1701.pdf.

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

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)*. Association for Computational Linguistics, Berlin, Germany, pages 1181–1191. http://www.aclweb.org/anthology/P16-1112.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The illinois-columbia system in the conll-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 34–42. http://www.aclweb.org/anthology/W14-1704.

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*. Association for Computational Linguistics, Berlin, Germany, pages 2205–2215. http://aclweb.org/anthology/P16-1208.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics* 4:169–182. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/800.

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)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 366–372. http://www.aclweb.org/anthology/I17-2062.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 951–962. http://www.aclweb.org/anthology/D14-1102.

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

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*. Association for Computational Linguistics, Portland, Oregon, USA, pages 180–189. http://www.aclweb.org/anthology/P11-1019.

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*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2795–2806. https://www.aclweb.org/anthology/D17-1297.

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*. Association for Computational Linguistics, San Diego, California, pages 380–386. http://www.aclweb.org/anthology/N16-1042.

Zheng Yuan, Ted Briscoe, and Mariano Felice. 2016. Candidate re-ranking for smt-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, San Diego, CA, pages 256–266. http://www.aclweb.org/anthology/W16-0530.