Adapting Sequence Models for Sentence Correction

Allen Schmaltz[*] **Yoon Kim** **Alexander M. Rush** **Stuart M. Shieber**

Harvard University

{schmaltz@fas,yoonkim@seas,srush@seas,shieber@seas}.harvard.edu

[*]Part of this work was completed while as an intern at Rakuten.

Abstract

In a controlled experiment of sequence-to-sequence approaches for the task of sentence correction, we find that character-based models are generally more effective than word-based models and models that encode subword information via convolutions, and that modeling the output data as a series of diffs improves effectiveness over standard approaches. Our strongest sequence-to-sequence model improves over our strongest phrase-based statistical machine translation model, with access to the same data, by 6 M^2 (0.5 GLEU) points. Additionally, in the data environment of the standard CoNLL-2014 setup, we demonstrate that modeling (and tuning against) diffs yields similar or better M^2 scores with simpler models and/or significantly less data than previous sequence-to-sequence approaches.

1 Introduction

The task of *sentence correction* is to convert a natural language sentence that may or may not have errors into a corrected version. The task is envisioned as a component of a learning tool or writing-assistant, and has seen increased interest since 2011 driven by a series of shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014).

Most recent work on language correction has focused on the data provided by the CoNLL-2014 shared task (Ng et al., 2014), a set of corrected essays by second-language learners. The CoNLL-2014 data consists of only around 60,000 sentences, and as such, competitive systems have made use of large amounts of corrected text without annotations, and in some cases lower-quality crowd-annotated data, in addition to the shared data. In this data environment, it has been suggested that statistical phrase-based machine translation (MT) with task-specific features is the state-of-the-art for the task (Junczys-Dowmunt and Grundkiewicz, 2016), outperforming word- and character-based sequence-to-sequence models (Yuan and Briscoe, 2016; Xie et al., 2016; Ji et al., 2017), phrase-based systems with neural features (Chollampatt et al., 2016b,a), re-ranking output from phrase-based systems (Hoang et al., 2016), and combining phrase-based systems with classifiers trained for hand-picked subsets of errors (Rozovskaya and Roth, 2016).

We revisit the comparison across translation approaches for the correction task in light of the Automated Evaluation of Scientific Writing (AESW) 2016 dataset, a correction dataset containing over 1 million sentences, holding constant the training data across approaches. The dataset was previously proposed for the distinct binary classification task of grammatical error identification.

Experiments demonstrate that pure character-level sequence-to-sequence models are more effective on AESW than word-based models and models that encode subword information via convolutions over characters, and that representing the output data as a series of *diffs* significantly increases effectiveness on this task. Our strongest character-level model achieves statistically significant improvements over our strongest phrase-based statistical machine translation model by 6 M^2 (0.5 GLEU) points, with additional gains when including domain information. Furthermore, in the partially crowd-sourced data environment of the standard CoNLL-2014 setup in which there are comparatively few professionally annotated sentences, we find that tuning against the tags marking the diffs yields similar or superior effectiveness relative to existing sequence-to-sequence approaches despite using significantly less data, with or without using secondary models. All code is available at https://github.com/allenschmaltz/grammar.

2 Background and Methods

Task We follow recent work and treat the task of sentence correction as translation from a source sentence (the unedited sentence) into a target sentence (a corrected version in the same language as the source). We do not make a distinction between grammatical and stylistic corrections.

We assume a vocabulary $\mathcal{V}$ of natural language word types (some of which have orthographic errors). Given a sentence $\mathbf{s} = [s_1 \cdots s_I]$, where $s_i \in \mathcal{V}$ is the i-th token of the sentence of length I, we seek to predict the corrected target sentence $\mathbf{t} = [t_1 \cdots t_J]$, where $t_j \in \mathcal{V}$ is the j-th token of the corrected sentence of length J. We are given both $\mathbf{s}$ and $\mathbf{t}$ for supervised training in the standard setup. At test time, we are only given access to sequence $\mathbf{s}$. We learn to predict sequence $\mathbf{t}$ (which is often identical to $\mathbf{s}$).

Sequence-to-sequence We explore word and character variants of the sequence-to-sequence framework. We use a standard word-based model (WORD), similar to that of Luong et al. (2015), as well as a model that uses a convolutional neural network (CNN) and a highway network over characters (CHARCNN), based on the work of Kim et al. (2016), instead of word embeddings as the input to the encoder and decoder. With both of these models, predictions are made at the word level. We also consider the use of bidirectional versions of these encoders (+BI).

Our character-based model (CHAR+BI) follows the architecture of the WORD+BI model, but the input and output consist of characters rather than words. In this case, the input and output sequences are converted to a series of characters and whitespace delimiters. The output sequence is converted back to $\mathbf{t}$ prior to evaluation.

The WORD models encode and decode over a closed vocabulary (of the 50k most frequent words); the CHARCNN models encode over an open vocabulary and decode over a closed vocabulary; and the CHAR models encode and decode over an open vocabulary.

Our contribution is to investigate the impact of sequence-to-sequence approaches (including those not considered in previous work) in a series of controlled experiments, holding the data constant. In doing so, we demonstrate that on a large, professionally annotated dataset, the most effective sequence-to-sequence approach can significantly outperform a state-of-the-art SMT system without augmenting the sequence-to-sequence model with a secondary model to handle low-frequency words (Yuan and Briscoe, 2016) or an additional model to improve precision or intersecting a large language model (Xie et al., 2016). We also demonstrate improvements over these previous sequence-to-sequence approaches on the CoNLL-2014 data and competitive results with Ji et al. (2017), despite using significantly less data.

The work of Schmaltz et al. (2016) applies WORD and CHARCNN models to the distinct binary classification task of error identification.

Additional Approaches The standard formulation of the correction task is to model the output sequence as $\mathbf{t}$ above. Here, we also propose modeling the diffs between $\mathbf{s}$ and $\mathbf{t}$. The diffs are provided in-line within $\mathbf{t}$ and are described via tags marking the starts and ends of insertions and deletions, with replacements represented as deletion-insertion pairs, as in the following example selected from the training set: "Some key points are worth <del> emphasiz </del> <ins> emphasizing </ins> .". Here, "emphasiz" is replaced with "emphasizing". The models, including the CHAR model, treat each tag as a single, atomic token.

The diffs enable a means of tuning the model's propensity to generate corrections by modifying the probabilities generated by the decoder for the 4 diff tags, which we examine with the CoNLL data. We include four bias parameters associated with each diff tag, and run a grid search between 0 and 1.0 to set their values based on the tuning set.

It is possible for models with diffs to output invalid target sequences (for example, inserting a word without using a diff tag). To fix this, a deterministic post-processing step is performed (greedily from left to right) that returns to source any non-source tokens outside of insertion tags. Diffs are removed prior to evaluation. We indicate models that *do not* incorporate target diff annotation tags with the designator –DIFFS.

The AESW dataset provides the paragraph context and a journal domain (a classification of the document into one of nine subject categories) for each sentence.[1] For the sequence-to-sequence

[1] The paragraphs are shuffled for purposes of obfuscation,

Model	GLEU		M^2	
	Dev	Test	Dev	Test
No Change	89.68	89.45	00.00	00.00
SMT–diffs+M^2	90.44	–	38.55	–
SMT–diffs+BLEU	90.90	–	37.66	–
Word+Bi–diffs	91.18	–	38.88	–
Char+Bi–diffs	91.28	–	40.11	–
SMT+BLEU	90.95	90.70	38.99	38.31
Word+Bi	91.34	91.05	43.61	42.78
CharCNN	91.23	90.96	42.02	41.21
Char+Bi	**91.46**	**91.22**	**44.67**	**44.62**
Word+dom	91.25	–	43.12	–
Word+Bi+dom	91.45	–	44.33	–
CharCNN+Bi+dom	91.15	–	40.79	–
CharCNN+dom	91.35	–	43.94	–
Char+Bi+dom	**91.64**	**91.39**	**47.25**	**46.72**

Table 1: AESW development/test set correction results. GLEU and M^2 differences on test are statistically significant via paired bootstrap resampling (Koehn, 2004; Graham et al., 2014) at the 0.05 level, resampling the full set 50 times.

models we propose modeling the input and output sequences with a special initial token representing the journal domain (+dom).[2]

3 Experiments

Data AESW (Daudaravicius, 2016; Daudaravicius et al., 2016) consists of sentences taken from academic articles annotated with corrections by professional editors used for the AESW shared task. The training set contains 1,182,491 sentences, of which 460,901 sentences have edits. We set aside a 9,947 sentence sample from the original development set for tuning (of which 3,797 contain edits), and use the remaining 137,446 sentences as the dev set[3] (of which 53,502 contain edits). The test set contains 146,478 sentences.

The primary focus of the present study is conducting controlled experiments on the AESW dataset, but we also investigate results on the CoNLL-2014 shared task data in light of recent neural results (Ji et al., 2017) and to serve as a baseline of comparison against existing sequence-to-sequence approaches (Yuan and Briscoe, 2016; Xie et al., 2016). We use the common sets of public data appearing in past work for training: the National University of Singapore (NUS) Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) and the publicly available Lang-8 data (Tajiri et al., 2012; Mizumoto et al., 2012). The Lang-8 dataset of corrections is large[4] but is crowd-sourced[5] and is thus of a different nature than the professionally annotated AESW and NUCLE datasets. We use the revised CoNLL-2013 test set as a tuning/dev set and the CoNLL-2014 test set (without alternatives) for testing. We do not make use of the non-public Cambridge Learner Corpus (CLC) (Nicholls, 2003), which contains over 1.5 million sentence pairs.

Evaluation We follow past work and use the Generalized Language Understanding Evaluation (GLEU) (Napoles et al., 2016) and MaxMatch (M^2) metrics (Dahlmeier and Ng, 2012).

Parameters All our models, implemented with OpenNMT (Klein et al.), are 2-layer LSTMs with 750 hidden units. For the Word model, the word embedding size is also set to 750, while for the CharCNN and Char models we use a character embedding size of 25. The CharCNN model has a convolutional layer with 1000 filters of width 6 followed by max-pooling, which is fed into a 2-layer highway network. Additional training details are provided in Appendix A. For AESW, the Word+Bi model contains around 144 million parameters, the CharCNN+Bi model around 79 million parameters, and the Char+Bi model around 25 million parameters.

Statistical Machine Translation As a baseline of comparison, we experiment with a phrase-based machine translation approach (SMT) shown to be state-of-the-art for the CoNLL-2014 shared task data in previous work (Junczys-Dowmunt and Grundkiewicz, 2016), which adds task specific features and the M^2 metric as a scorer to the Moses statistical machine translation system. The SMT model follows the training, parameters, and dense and sparse task-specific features that generate state-of-the-art results for CoNLL-2014 shared task data, as implemented in publicly available code.[6] However, to compare models against the same training data, we remove language model features associated with external data.[7] We exper-

so document-level context is not available.

[2] Characteristics of the dataset preclude experiments with additional paragraph context features. (See Appendix A.)

[3] The dev set contains 13,562 unique deletion types, 29,952 insertion types, and 39,930 replacement types.

[4] about 1.4 million sentences after filtering

[5] derived from the Lang-8 language-learning website

[6] SRI International provided access to SRILM (Stolcke, 2002) for running Junczys-Dowmunt and Grundkiewicz (2016)

[7] We found that including the features and data associated with the large language models of Junczys-Dowmunt and Grundkiewicz (2016), created from Common Crawl text

iment with tuning against M^2 (+M^2) and BLEU (+BLEU). Models trained with diffs were only tuned with BLEU, since the tuning pipeline from previous work is not designed to handle removing such annotation tags prior to M^2 scoring.

4 Results and Analysis: AESW

Table 1 shows the full set of experimental results on the AESW development and test data.

The CHAR+BI+DOM model is stronger than the WORD+BI+DOM and CHARCNN+DOM models by 2.9 M^2 (0.2 GLEU) and 3.3 M^2 (0.3 GLEU), respectively. The sequence-to-sequence models were also more effective than the SMT models, as shown in Table 1. We find that training with target diffs is beneficial across all models, with an increase of about 5 M^2 points for the WORD+BI model, for example. Adding +DOM information slightly improves effectiveness across models.

We analyzed deletion, insertion, and replacement error types. Table 2 compares effectiveness across replacement errors. We found the CHARCNN+BI models were less effective than CHARCNN variants in terms of GLEU and M^2, and the strongest CHARCNN models were eclipsed by the WORD+BI models in terms of the GLEU and M^2 scores. However, Table 2 shows CHARCNN+DOM is stronger on lower frequency replacements than WORD models. The CHAR+BI+DOM model is relatively strong on article and punctuation replacements, as well as errors appearing with low frequency in the training set and overall across deletion and insertion error types, which are summarized in Table 3.

Errors never occurring in training The comparatively high Micro $F_{0.5}$ score (18.66) for the CHAR+BI+DOM model on replacement errors (Table 2) never occurring in training is a result of a high precision (92.65) coupled with a low recall (4.45). This suggests some limited capacity to generalize to items not seen in training. A selectively chosen example is the replacement from "discontinous" to "discontinuous", which never occurs in training. However, similar errors of low edit distance also occur once in the dev set and never in training, but the CHAR+BI+DOM model never correctly recovers many of these errors, and many of the correctly recovered errors are minor changes in capitalization or hyphenation.

Error frequency About 39% of the AESW training sentences have errors, and of those sentences, on average, 2.4 words are involved in changes in deletions, insertions, or replacements (i.e., the count of words occurring between diff tags) per sentence. In the NUCLE data, about 37% of the sentences have errors, of which on average, 5.3 words are involved in changes. On the AESW dev set, if we only consider the 9545 sentences in which 4 or more words are involved in a change (average of 5.8 words in changes per sentence), the CHAR+BI model is still more effective than SMT+BLEU, with a GLEU score of 67.21 vs. 65.34. The baseline GLEU score (No Change) is 60.86, reflecting the greater number of changes relative to the full dataset (cf. Table 1).

Re-annotation The AESW dataset only provides 1 annotation for each sentence, so we perform a small re-annotation of the data to gauge effectiveness in the presence of multiple annotations. We collected 3 outputs (source, gold, and generated sentences from the CHAR+BI+DOM model) for 200 randomly sampled sentences, reannotating to create 3 new references for each sentence. The GLEU scores for the 200 original source, CHAR+BI+DOM, and original gold sentences evaluated against the 3 new references were 79.79, 81.72, and 84.78, respectively, suggesting that there is still progress to be made on the task relative to human levels of annotation.

5 Results and Analysis: CoNLL

Table 4 shows the results on the CoNLL dev set, and Table 5 contains the final test results.

Since the CoNLL data does not contain enough data for training neural models, previous works add the crowd-sourced Lang-8 data; however, this data is not professionally annotated. Since the distribution of corrections differs between the dev/test and training sets, we need to tune the precision and recall.

As shown in Table 4, WORD+BI effectiveness increases significantly by tuning the weights[8] assigned to the diff tags on the CoNLL-2013 set[9].

filtered against the NUCLE corpus, *hurt* effectiveness for the phrase-based models. This is likely a reflection of the domain specific nature of the academic text and LaTeX holder symbols appearing in the text. Here, we conduct controlled experiments without introducing additional domain-specific monolingual data.

[8] In contrast, in early experiments on AESW, tuning yielded negligible improvements.

[9] The single model with highest M^2 score was then run on the test set. Here, a single set is used for tuning and dev.

Replacement Error Type (out of 39,930) – Frequency relative to training							
Model	Punctuation	Articles	Other				
			> 100	$[5, 100]$	$[2, 5)$	1	0
Raw frequency in dev	11507	1691	6788	8974	2271	1620	7079
Number of unique instances	371	367	215	2918	1510	1242	5819
SMT+BLEU	56.03	16.41	44.57	36.17	39.46	31.93	0.00
WORD+BI	56.13	18.58	55.38	44.33	18.79	6.38	0.77
WORD+BI+DOM	56.87	19.16	59.02	44.57	19.70	4.42	2.01
CHARCNN+DOM	55.64	13.37	57.34	41.83	28.99	16.74	7.09
CHAR+BI	58.71	28.40	55.34	44.59	28.98	24.48	14.14
CHAR+BI+DOM	58.93	27.64	59.32	46.08	32.82	26.48	18.66

Table 2: Micro $F_{0.5}$ scores on replacement errors on the dev set. Errors are grouped by 'Punctuation', 'Article', and 'Other'. 'Other' errors are further broken down based on frequency buckets on the training set, with errors grouped by the frequency in which they occur in the training set.

	Deletions	Insertions	Replacements
SMT+BLEU	46.56	31.48	42.21
WORD+BI	47.75	38.31	46.02
WORD+BI+DOM	47.78	39.00	47.29
CHARCNN+DOM	48.30	39.57	46.24
CHAR+BI	49.05	37.17	48.55
CHAR+BI+DOM	50.20	42.51	50.39

Table 3: Micro $F_{0.5}$ scores across error types

	Precision	Recall	$F_{0.5}$
WORD+BI–DIFFS	65.36	6.19	22.45
WORD+BI, before tuning	72.34	0.97	4.60
WORD+BI, after tuning	46.66	15.35	33.14

Table 4: M^2 scores on the CoNLL-2013 set.

Note that we are tuning the weights on this same CoNLL-2013 set. Without tuning, the model very rarely generates a change, albeit with a high precision. After tuning, it exceeds the effectiveness of WORD+BI–DIFFS. The comparatively low effectiveness of WORD+BI–DIFFS is consistent with past sequence-to-sequence approaches utilizing data augmentation, additional annotated data, and/or secondary models to achieve competitive levels of effectiveness.

Table 5 shows that WORD+BI is within 0.2 M^2 of Ji et al. (2017), despite using over 1 million fewer sentence pairs, and exceeds the M^2 scores of Xie et al. (2016) and Yuan and Briscoe (2016) without the secondary models of those systems. We hypothesize that further gains are possible utilizing the CLC data and moving to the character model. (The character model is omitted here due to the long training time of about 4 weeks.) Notably, SMT systems (with LMs) are still more effective than reported sequence-to-sequence results, as in Ji et al. (2017), on CoNLL.[10]

	Data	M^2
Yuan and Briscoe (2016)	CLC*	39.90
Xie et al. (2016)	NUCLE, Lang-8, Common Crawl LM	40.56
Ji et al. (2017)	NUCLE, Lang-8, CLC*	41.53
WORD+BI–DIFFS	NUCLE, Lang-8	35.73
WORD+BI	NUCLE, Lang-8	41.37

Table 5: M^2 scores on the CoNLL-2014 test set and data used for recent sequence-to-sequence based systems. Results for previous works are those reported by the original authors. *CLC is proprietary.

6 Conclusion

Our experiments demonstrate that on a large, professionally annotated dataset, a sequence-to-sequence character-based model of diffs can lead to considerable effectiveness gains over a state-of-the-art SMT system with task-specific features, ceteris paribus. Furthermore, in the crowd-sourced environment of the CoNLL data, in which there are comparatively few professionally annotated sentences in training, modeling diffs enables a means of tuning that improves the effectiveness of sequence-to-sequence models for the task.

[10] For reference, the reported M^2 results of the carefully optimized SMT system of Junczys-Dowmunt and Grundkiewicz (2016) trained on NUCLE and Lang-8, with parameter vectors averaged over multiple runs, with a Wikipedia LM is 45.95 and adding a Common Crawl LM is 49.49. We leave to future work the intersection of a LM for the CoNLL environment and more generally, whether these patterns hold in the presence of additional monolingual data.

References

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016a. 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*, pages 1901–1911, Austin, Texas. Association for Computational Linguistics.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016b. Neural network translation models for grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2768–2774. AAAI Press.

Daniel Dahlmeier and Hwee Tou Ng. 2012. 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*, NAACL HLT '12, pages 568–572, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 242–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vidas Daudaravicius. 2016. Automated evaluation of scientific writing data set (version 1.2) [data file]. VTeX.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can markov models over minimal translation units help phrase-based smt? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 399–405, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014. Randomized significance tests in machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 266–274, Baltimore, Maryland, USA. Association for Computational Linguistics.

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, pages 2803–2809. AAAI Press.

J. Ji, Q. Wang, K. Toutanova, Y. Gong, S. Truong, and J. Gao. 2017. A Nested Attention Neural Hybrid Model for Grammatical Error Correction. *ArXiv e-prints*.

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.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *Proceedings of AAAI*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012: Posters*, pages 863–872, Mumbai, India. The COLING 2012 Organizing Committee.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. GLEU without tuning. *eprint arXiv:1605.02592 [cs.CL]*.

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.

Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 572–581.

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.

Allen Schmaltz, Yoon Kim, Alexander M. Rush, and Stuart Shieber. 2016. Sentence-level grammatical error identification as sequence-to-sequence correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 242–251, San Diego, CA. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904, Denver.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. 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.

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.

A Supplemental Material

Additional Model Training and Inference Details We provide additional replication details for our experiments here. Our code and related materials are available at the following url: `https://github.com/allenschmaltz/grammar`.

The training and tuning sizes of the AESW dataset are those after dropping sentences exceeding 126 tokens on the source or target side (in source sequences or target sequences with diff annotation tags) from the raw AESW dataset. All evaluation metrics on the development and test set are on the data without filtering based on sentence lengths.

As part of preprocessing, the sentences from the AESW XML are converted to Penn Treebank-style tokenization. Case is maintained and digits are not replaced with holder symbols for the sequence-to-sequence models. For the SMT models, the truecasing[11] and tokenization pipeline of the publicly available code is used. For consistency, all model output and all reference files are converted to cased Moses-style tokenization prior to evaluation.

For the CHAR model, the L_2-normalized gradients were constrained to be ≤ 1 (instead of ≤ 5 with the other models), and our learning rate schedule started the learning rate at 0.5 (instead of 1 for the other models) for stable training. The maximum sequence length of 421 was used for models given character sequences, which was equivalent to the maximum sequence length of 126 used for models given word sequences. The maximum sequence lengths were increased by 1 for the models with the +DOM features. The training and tuning set sizes cited in Section 3 are the number of sentences from the raw dataset after dropping sentences exceeding these maximum sequence lengths.

In practice, we were able to train each of the purely character-based models (e.g., the CHAR+BI+DOM model) with a single NVIDIA Quadro P6000 GPU with 24 GB of memory in about 3 weeks with a batch size of 12.

For the sequence-to-sequence models, the closed vocabularies were restricted to the 50,000 most common tokens, and a single special <unk> token was used for all remaining low frequency tokens. An <unk> token generated in the target sentence by the WORD and CHARCNN models was replaced with the source token associated with the maximum attention weight. The "open" vocabularies were only limited to the space of char-

[11] Here, the truecase language model is created from the training $\mathbf{t}$ sequences (or where applicable, the target with diffs).

acters seen in training.

For the phrase-based machine translation baseline model from the work of Junczys-Dowmunt and Grundkiewicz (2016), for dense features, we used the stateless edit distance features and the stateful Operation Sequence Model (OSM) of Durrani et al. (2013)[12]. Since for our controlled data experiments we removed the language model features associated with external data, we did not use the word-class language model feature, so for the sparse features, we used the set of edit operations on "words with left/right context of maximum length 1 on words" (set "E0C10" from the original paper), instead of those dependent on word classes.

The training and tuning splits for the phrase-based machine translation models were the same as for the sequence-to-sequence models. For tuning, we used Batch-Mira, setting the background corpus decay rate to 0.001, as in previous work. As in previous work, we repeated the tuning process multiple times (in this case, 5 times) and averaged the final weight vectors.

The sequence-to-sequence models were decoded with a beam size of 10.

Decoding of the SMT models used the same approach of Junczys-Dowmunt and Grundkiewicz (2016) (i.e., the open-source Moses decoder run with the cube pruning search algorithm).

In our experiments, we do not include additional paragraph context features, since the underlying AESW data appears to have been collected such that nearly all paragraphs (including those containing a single sentence) contain at least one error; thus, modeling paragraph information provides additional signal that seems unlikely to reflect real-world environments.

CoNLL-2014 Shared Task For training, we used the copy of the Lang-8 corpus distributed in the repo for the code of Junczys-Dowmunt and Grundkiewicz (2016): `https://github.com/grammatical/baselines-emnlp2016`. We filtered the Lang-8 data to remove duplicates and target sentences containing emoticon text, informal colloquial words (e.g., "haha", "lol", "yay"), and non-ascii characters. Target sentences not starting with a capital letter were dropped, as were target sentences not ending in a period, question mark, exclamation mark, or quotation mark. (Target sentences ending in a parenthesis were dropped as they often indicate informal additional comments from the editor.) In the combined NUCLE and Lang-8 training set, source sentences longer than 79 tokens and target sentences longer than 100 tokens were dropped. This resulted in a training set with 1,470,992 sentences. Diffs were created using the Python class difflib.SequenceMatcher.

Bias parameter	Precision	Recall	$F_{0.5}$
0.0	72.34	0.97	4.60
0.1	69.74	1.51	6.96
0.2	72.00	2.57	11.23
0.3	69.05	4.14	16.68
0.4	67.19	6.08	22.31
0.5	61.03	8.76	27.82
0.6	51.75	11.41	30.31
0.7	**46.66**	**15.35**	**33.14**
0.8	40.01	18.68	32.57
0.9	34.49	22.08	31.00
1.0	30.17	24.90	28.94

Table 6: M^2 scores on the CoNLL-2013 dev set for the WORD+BI model.

For tuning on the dev set[13], a coarse grid search between 0 and 1.0 was used to set the four bias parameters associated with each diff tag. (Training was performed without re-weighting.) The bias parameter (in this case 0.7) yielding the highest M^2 score on the decoded dev set was chosen for use in evaluation of the final test set. The M^2 scores across the tuning runs on the dev set for the WORD+BI model are shown in Table 6.

For future comparisons to our work on the CoNLL-2014 shared task data, we recommend using the preprocessing scripts provided in our code repo (`https://github.com/allenschmaltz/grammar`).

Table 2 The seven columns of Table 2 appearing in the main text are Micro $F_{0.5}$ scores for the errors within each frequency grouping. There are a total of 39,916 replacement changes. The replacements are grouped in regard to the changes within the opening and closing deletion tags and subsequent opening and closing insertion tags, as follows: (1) whether the replacement involves (on the deletion and/or insertion side) a single punctuation symbol (comma, colon, period, hyphen, apostrophe, quotation mark, semicolon, exclamation, question

[12] The OSM features use the SRI Language Modeling Toolkit (SRILM) (Stolcke, 2002).

[13] Previous work, such as Junczys-Dowmunt and Grundkiewicz (2016), also used the CoNLL-2013 set for tuning.

mark); (2) whether the replacement involves (on the deletion and/or insertion side) a single article (a, an, the); (3) non-article, non-punctuation grouped errors with frequency greater than 100 in the gold training data; (4) non-article, non-punctuation grouped errors with frequency less than or equal to 100 and greater than or equal to 5; (5) non-article, non-punctuation grouped errors with frequency less than 5 and greater than or equal to 2; (6) non-article, non-punctuation grouped errors with frequency equal to 1; (7) non-article, non-punctuation grouped errors that never occurred in the training data. Note that the large number of unique instances occurring for the "punctuation" and "articles" classes are a result of the large number of errors that can occur on the non-article, non-punctuation side of the replacement. The Micro $F_{0.5}$ scores are calculated by treating each individual error (rather than the agglomerated classes here) as binary classifications.