Document-level grammatical error correction

Zheng Yuan and **Christopher Bryant**
ALTA Institute, University of Cambridge, United Kingdom
Department of Computer Science and Technology, University of Cambridge, United Kingdom
{zheng.yuan, christopher.bryant}@cl.cam.ac.uk

Abstract

Document-level context can provide valuable information in grammatical error correction (GEC), which is crucial for correcting certain errors and resolving inconsistencies. In this paper, we investigate context-aware approaches and propose document-level GEC systems. Additionally, we employ a three-step training strategy to benefit from both sentence-level and document-level data. Our system outperforms previous document-level and all other NMT-based single-model systems, achieving state of the art on a common test set.

1 Introduction

Grammatical error correction (GEC) attempts to automatically detect and correct grammatical errors in text. With recent advances in sequence-to-sequence modelling, neural machine translation (NMT) has been widely applied to GEC (Yuan and Briscoe, 2016; Ji et al., 2017; Junczys-Dowmunt et al., 2018; Yuan et al., 2019) and state-of-the-art results have been reported (Kaneko et al., 2020; Lichtarge et al., 2020). Cross-sentence context has proven useful for language modelling (Wang and Cho, 2016), dialogue systems (Serban et al., 2016) and machine translation (Wang et al., 2017; Voita et al., 2018). In error correction, we observe that certain errors can only be detected and/or corrected using wider context, which may fall outside the current sentence. However, existing GEC systems typically process each sentence independently, ignoring document-level context. These sentence-level systems may fail to correct document-level errors (e.g. verb tense errors, pronoun errors, run-on sentences) or propose inconsistent corrections throughout a document:

(a) In the chat room, she created a close relationship with eight people. She talks (**talked**) to them every night, trust (**trusted / ~~trusts~~**) them and share (**shared / ~~shares~~**) her life with them. Then eventually, she discovered that the eight people were one as the other person was using eight different identities to chat with her all the time.

(b) I would like to recommend walking. Because there are a lot of beautiful trees. → I would like to recommend walking **because** there are a lot of beautiful trees.

For example, all the errors (red) in Example (a) could feasibly be corrected (bold) using either the present or past tense if we only consider the target sentence in isolation, but the wider context reveals the correction using the present tense is ungrammatical (strikethrough). Similarly, a sentence-level system is also unable to handle cases where sentences should be merged such as in Example (b).

To date, GEC evaluation has always been carried out at the sentence level. As a result, successful corrections of these document-level errors would not be given any credit (or even be unfairly penalised) and systems proposing inconsistent modifications would not be penalised. On the one hand, GEC should look beyond the current sentence and use more context to build context-aware GEC systems; on the other hand, systems should be better evaluated at the document rather than sentence level.

This paper makes the following contributions. First, we compare different architectures to capture wider context for NMT-based GEC and show that simple document-level approaches can be applied to improve GEC performance. Second, we present a three-step training strategy to effectively use both sentence-level and document-level parallel data for GEC. Third, we report state of the art on a publicly available test set. Finally, we perform the first document-level GEC evaluation and release

Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 75–84
April 20, 2021

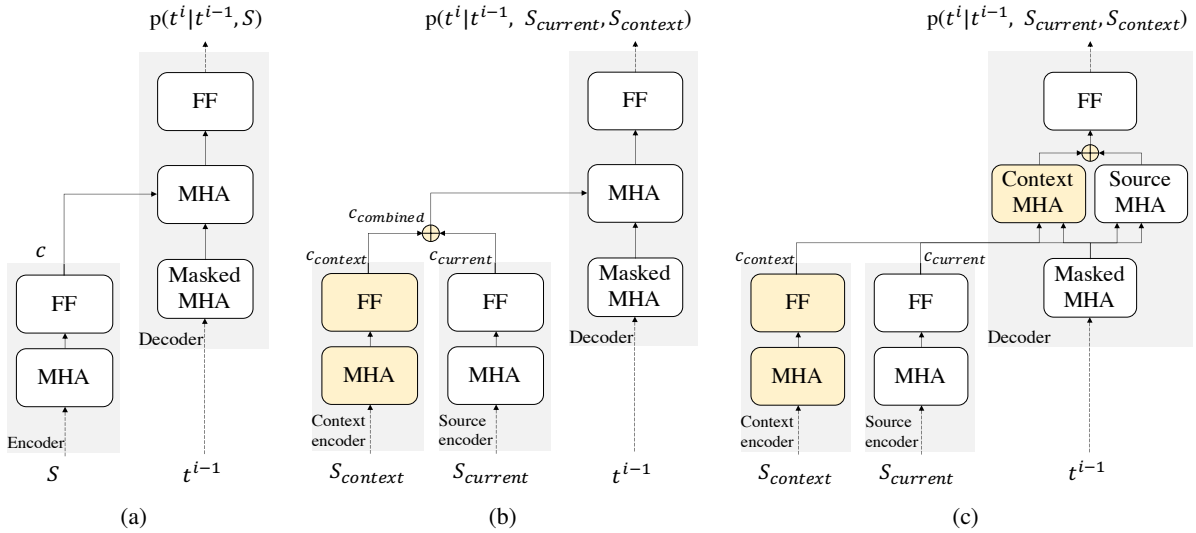

Figure 1: (a) The original Transformer, (b) the multi-encoder model with encoder side integration (**MultiEnc-enc**), and (c) the multi-encoder model with decoder side integration (**MultiEnc-dec**). The newly introduced components are highlighted in yellow. FF: Feed Forward, MHA: Multi-Head Attention.

our document-level evaluation scripts to facilitate research in the area.[1]

2 Document-level GEC

NMT was originally developed to work sentence by sentence (Sutskever et al., 2014). Recent work has explored context-aware extensions. A simple strategy of concatenating preceding sentences was investigated by Tiedemann and Scherrer (2017). Multi-encoder approaches have been proposed, including using different encoder architectures (Bawden et al., 2018; Chollampatt et al., 2019; Stojanovski and Fraser, 2018), and applying multiple integration strategies (Wang et al., 2017; Voita et al., 2018; Bawden et al., 2018).

Various adaptations of NMT for GEC have been investigated and recent progress has been driven by the use of artificial data (Kaneko et al., 2020; Lichtarge et al., 2020). However most systems focus on sentence-level correction, where each sentence is processed in isolation. The only previous work that has considered a wider context for GEC that we are aware of is by Chollampatt et al. (2019), who extended a convolutional neural encoder-decoder model with an auxiliary encoder and attention gating. They only used document-level data in their training however, and still performed evaluation at the sentence level, which is therefore unable to ascertain the real improvements of document-level systems.

[1] https://github.com/chrisjbryant/doc-gec

In this work, we use the Transformer sequence-to-sequence model (Vaswani et al., 2017) as our baseline system and investigate three context-aware extensions for GEC.

2.1 Baseline encoder-decoder framework

The Transformer follows an encoder-decoder architecture (Figure 1a). Each layer of the encoder contains a multi-head self-attention mechanism and a feed-forward network. The decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

2.2 Single-encoder models

Similar to Tiedemann and Scherrer (2017), the single-encoder GEC model uses the standard Transformer encoder to process the current source sentence and its context together, treating them as a long input sequence. The model architecture remains unchanged. We instead modify the input by concatenating preceding sentences to the current one, separated by a special token.

2.3 Multi-encoder models

Multi-encoder models encode the source sentence and its context separately. The original encoder in the Transformer reads and encodes the source sentence. Additionally, a new context encoder is introduced to process the context in a parallel fashion to the source encoder. The resulting context representation is integrated into the model architecture on the encoder or decoder side:

Encoder side integration As shown in Figure 1b, the original source encoder reads the current source sentence $S_{current}$ and produces a vector representation $c_{current}$. The context encoder encodes the auxiliary context input $S_{context}$ and computes a context representation $c_{context}$. The outputs from both encoders are combined via a gated sum:

$$c_{combined} = \lambda c_{current} + (1 - \lambda) c_{context} \quad (1)$$

where the gating weight λ is given by:

$$\lambda = \sigma(W [c_{current}; c_{context}] + b) \quad (2)$$

where σ is the logistic sigmoid function, and W and b are learnable parameters.

The combined representation $c_{combined}$ is then used as a single input to the decoder which stays intact.

Decoder side integration Two encoder representations $c_{current}$ and $c_{context}$ are used as separate inputs to the decoder (Figure 1c). We modify the Transformer decoder, so that the multi-head attention sub-layer contains two components: one performs multi-head attention over the output of the encoder stack for the current sentence $c_{current}$ using the masked multi-head self-attention output, and the other attends directly to the context encoder representation $c_{context}$. These two attention operations are performed in parallel and combined with a gating mechanism.

3 Experiments

3.1 Datasets

To train document-level GEC models, we select document-level corpora: the Cambridge English Write & Improve (W&I) corpus (Bryant et al., 2019), the First Certificate in English (FCE) dataset (Yannakoudakis et al., 2011), the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) and the Cambridge Learner Corpus (CLC) (Nicholls, 2003).[2] All the datasets were annotated by expert annotators at the document level and typically consist of learner essays. We use FCE-dev as our development set and report results on FCE-test, BEA-dev (Granger, 1998; Bryant et al., 2019) and the CoNLL-2014 test set (Ng et al., 2014).[3] More information about these datasets is provided in Appendix A, Table A.1.

3.2 Document-level GEC evaluation

To better understand the performance of document-level systems, we perform the first document-level GEC evaluation. We do this using the ERRANT Scorer (Bryant et al., 2017), the official scorer of the BEA-2019 shared task (Bryant et al., 2019). Since reference files are normally only available at the sentence level, we reprocess the raw untokenised data to produce new reference files at the document level.[4] This is necessary because edits that cross sentence boundaries are normally deleted in sentence-level GEC (see Example (b) in Section 1). It is also worth noting that for datasets with multiple references (i.e. CoNLL-2014), scores are computed against all the document-level edits of a single annotator simultaneously rather than mixed-and-matched from different annotators for each sentence. In other words, while sentence-level evaluation chooses the best reference amongst all annotators *for each sentence*, document-level evaluation chooses the best reference amongst all annotators *for each document*. This means document-level evaluation is more restricted than sentence-level evaluation and hence explains why the document-level scores in our experiments on CoNLL-2014 are much lower than the sentence-level scores.

3.3 Training

The implementation is done using Fairseq, an open-source sequence modelling toolkit (Ott et al., 2019).[5] We use the Transformer model as the basic model architecture and follow the hyper-parameter settings in 'Transformer (big)' in Vaswani et al. (2017). We apply byte pair encoding (Sennrich et al., 2016) with 8k merge operations learned from the target side of the training data. Source word embeddings are shared between the source and context encoders.[6] In our experiments, one preceding source sentence is given as the context. Each model is trained on one machine with four NVIDIA Tesla P100 GPUs.

Since large-scale document-level GEC corpora are limited and existing methods for artificial error generation work at the sentence level (Felice and

[2] All public data is available at: https://www.cl.cam.ac.uk/research/nl/bea2019st/#data

[3] We do not use BEA-test or JFLEG (Napoles et al., 2017) because document-level context for these datasets is not available.

[4] This preprocessing was done using a modified version of the json_to_m2.py script released with the BEA-2019 shared task data.

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

[6] Detailed hyper-parameters are listed in Appendix B.

Model	BEA-dev			FCE-test			CoNLL-2014		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Baseline	58.49	38.29	52.91	63.65	42.27	57.80	59.96	27.08	48.25
SingleEnc	56.94	**43.16**	53.52	61.63	**44.95**	57.37	59.78	27.27	48.27
MultiEnc-enc	62.06	41.71	56.54	**65.55**	42.68	59.20	63.23	27.96	50.49
MultiEnc-dec	**62.64**	40.72	**56.55**	65.36	44.17	**59.64**	**64.57**	**28.65**	**51.62**

Table 1: Document-level evaluation results of our proposed document-level GEC models. The highest scores are marked in bold. P: precision, R: recall.

Stage	Data	P	R	$F_{0.5}$
Pre-training	sent.	57.52	32.65	49.91
Training	doc.	58.49	38.29	52.91
Fine-tuning	doc.	62.64	40.72	56.55
No pre-training	-	59.62	40.52	54.48
No fine-tuning	-	58.49	38.29	52.91

Table 2: Performance of **MultiEnc-dec** on BEA-dev after each training stage and ablation tests. sent.: sentence-level data, doc.: document-level data.

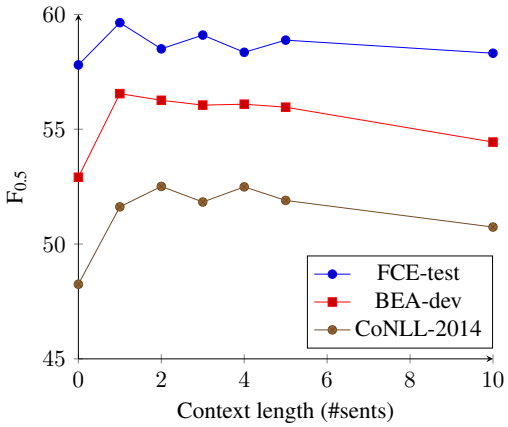

Figure 2: The effect of context length.

Yuan, 2014; Rei et al., 2017; Kiyono et al., 2019), we extract both sentence-level and document-level parallel training examples from the document-level GEC corpora. To train **MultiEnc-enc** and **MultiEnc-dec**, we employ a three-step training strategy: 1) pre-training on all sentence-level parallel data from *CLC* + *FCE-train* + *W&I-train* + *NUCLE* to learn sentence-level model parameters (the newly introduced components are therefore inactivated - see Figure 1b and 1c); 2) continue training with *CLC* document-level parallel data to update all model parameters; and 3) fine-tuning on a combination of small, in-domain document-level data from *FCE-train* + *W&I-train* + *NUCLE*. Both **Baseline** and **SingleEnc**, which follow the standard Transformer, are similarly first trained using *CLC*, then fine-tuned with in-domain *FCE-train* + *W&I-train* + *NUCLE* data, but without mixing both sentence-level and document-level examples.

3.4 Results

In Table 1, we can see that simply concatenating preceding sentences (**SingleEnc**) does not yield a consistent improvement in $F_{0.5}$ (recall improves at the cost of precision). Since longer inputs make the encoder-decoder attention harder to optimise, more training data may be needed. Both our document-level models outperform the sentence-level **Baseline**.[7] **MultiEnc-dec** gives the decoder more flexibility to access context directly, and produces better results on FCE-test and CoNLL-2014 than **MultiEnc-enc**, but makes no difference on BEA-dev. We also find that document-level context seems more useful in some datasets than others, which improves the **Baseline** by up to 3.64 $F_{0.5}$ on BEA-dev, 3.37 on CoNLL-2014, but just 1.84 on FCE-test.

Multi-stage training Results in Table 2 demonstrate the effectiveness of the three-step training strategy and the benefits of using both sentence-level and document-level data. The ablation study, in which we remove one training step at a time, also suggests that it is crucial to have both pre-training and fine-tuning stages as performance drops when removing either of them.

Context length Figure 2 shows how the performance changes in relation to an increasing number of context sentences. The best performance is achieved when including only one preceding sentence for FCE-test and BEA-dev, but two for CoNLL-2014. This could possibly be explained by the difference in document length in each dataset: CoNLL-2014 documents contain twice as many sentences on average than FCE-test and BEA-dev

[7] We perform two-tailed paired T-tests, where $p < 0.001$.

Example (a)		
	Context	Then we went to Taxco.
	Source	We stay in a very luxurious hotel.
	Reference	We **stayed** in a very luxurious hotel.
	Baseline	We stay in a very luxurious hotel.
	Our model	We **stayed** in a very luxurious hotel.
Example (b)		
	Context	The motorcycle is the most dangerous transport ...
	Source	... some riders still keep breaking the rule.
	Reference	... some riders still keep breaking the rule.
	Baseline	... some cyclists still keep breaking the rule.
	Our model	... some riders still keep breaking the rule.

Table 3: Example outputs from **MultiEnc-dec**. More system output examples are given in Appendix D.

System	FCE-test			CoNLL-2014		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
MultiEnc-dec	**69.9**	44.2	**62.6**	**74.3**	39.0	**62.9**
Chollampatt et al. (2019)	52.2	28.3	44.6	65.6	30.1	53.1
Kaneko et al. (2020)	65.0	49.6	61.2†	69.2	45.6	62.6
Lichtarge et al. (2020)	-	-	-	69.4	43.9	62.1

Table 4: Comparison of NMT-based single-model GEC systems. †current state of the art

documents. But we also notice that very long context is not often helpful in resolving many different kinds of grammatical errors, suggesting that long-distance context has limited impact on GEC.

3.5 Error analysis

Our error analysis shows that the biggest gains are observed for subject-verb agreement, preposition, noun number, determiner and pronoun errors.[8] This confirms our hypothesis that correction of errors involving agreement, coreference or tense is more likely to rely on information outside the current sentence (e.g. VERB:SVA +10.40 $F_{0.5}$, PRON +8.32, and VERB:TENSE +5.95 - see Example (a) in Table 3). It is not surprising that our system is good at handling errors that cross sentence boundaries (e.g. CONJ +6.40 and PUNCT +3.75). Manual inspection reveals that improvements also come from topic-aware lexical choice (e.g. 'riders' vs. 'cyclists' for 'motorcycle' - see Example (b) in Table 3).

4 Comparison with NMT-based GEC systems

We perform sentence-level evaluation on the FCE-test and CoNLL-2014 test sets using the M^2 Scorer (Dahlmeier and Ng, 2012). A comparison of NMT-based single model systems is made in Table 4. Our **MultiEnc-dec** system outperforms previous document-level GEC systems from Chollampatt et al. (2019) on both test sets by large margins. Our single-model system outperforms all NMT-based single-model systems and achieves state of the art on FCE-test without exploiting any artificial data. Our GEC system also yields much higher precision, which is a desirable property of a practical system. As the performance of our document-level system is underestimated by sentence-level evaluation, we expect further performance gains over other sentence-level systems.

5 Conclusion

We have investigated document-level approaches to NMT-based GEC and presented a three-step training strategy to use both sentence-level and document-level data. We have shown that context is useful in GEC but very long context is not necessary for improved performance. Experiments on three test sets demonstrated the effectiveness of our document-level GEC models. Our best system outperforms all NMT-based single-model GEC systems and achieves state of the art on FCE-test. By drawing attention to this understudied area in GEC, we hope to motivate future efforts to build better context-aware GEC systems. We have also performed the first document-level GEC evaluation and make our document-level evaluation scripts available to facilitate research in this area.

Acknowledgments

We would like to thank Cambridge Assessment for supporting this research, and the anonymous re-

[8] The full error type-specific performance is presented in Appendix C.

viewers for their useful feedback. We would also like to thank Shiva Taslimipoor, Christopher Davis, Andrew Caines, and Ted Briscoe for feedback on early drafts of this paper. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council. We acknowledge NVIDIA for an Academic Hardware Grant.

References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. 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 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. 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.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy. Association for Computational Linguistics.

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*, pages 568–572, Montréal, Canada. 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.

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

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

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, 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.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

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.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, Arizona, USA. Association for the Advancement of Artificial Intelligence.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1329, Berlin, Germany. 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.

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

Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.

A Corpus statistics

Split	Dataset	#tokens	#sents	#docs	Doc. length (avg. #sents)
Train	FCE-train	454,736	28,350	2,116	13
	W&I-train	628,720	34,308	3,000	11
	NUCLE	1,161,567	57,151	1,397	41
	CLC	28,988,729	1,961,065	206,418	10
Dev	FCE-dev	34,748	2,191	159	14
Test	FCE-test	41,932	2,695	194	14
	BEA-dev	86,973	4,384	350	13
	CoNLL-2014	30,144	1,312	50	26

Table A.1: Summary of datasets used in our experiments. All datasets were preprocessed using spaCy.[9]

B Hyper-parameter settings

Model architecture	Transformer (big)
Max tokens	3,584
Optimiser	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning rate schedule	Same as in Vaswani et al. (2017)
Loss function	Label smoothed cross entropy ($\epsilon_{ls} = 0.1$)
Dropout	0.3
Gradient clipping	1.0
Beam size	12

[9] https://spacy.io

C Error analysis

In order to better understand the performance of our document-level GEC systems, we perform a detailed error analysis on BEA-dev using the ERRANT Scorer (Table C.1). The largest improvements in $F_{0.5}$ over the sentence-level baseline are observed for VERB:SVA (+10.40), followed by PREP (+10.00), NOUN:NUM (+8.65), DET (+8.57), PRON (+8.32), CONJ (+6.40), VERB:TENSE (+5.95), VERB:FORM (+5.58) and PUNCT (+3.75). Results for NOUN:INFL (+31.25), VERB:INFL (+26.92), WO (+7.9) and ADJ:FORM (+6.94) are not highlighted because they are rare and only account for a small fraction of the data (0.05%, 0.08%, 1.16%, and 0.16% respectively).

Error type	**Sentence-level baseline**			**Document-level system**			**Diff. $F_{0.5}$**
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	
ADJ	42.55	15.62	31.65	44.44	15.62	32.47	+0.82
ADJ:FORM	66.67	33.33	55.56	100.00	25.00	62.50	+6.94
ADV	48.21	20.15	37.71	42.11	17.91	33.15	-4.56
CONJ	35.71	11.90	25.51	46.15	14.29	31.91	+6.40
CONTR	88.24	51.72	77.32	85.00	58.62	77.98	+0.66
DET	55.52	42.19	52.22	63.72	51.33	60.79	+8.57
MORPH	62.96	34.87	54.23	70.65	33.33	57.73	+3.50
NOUN	35.90	11.60	25.30	38.10	13.26	27.71	+2.41
NOUN:INFL	60.00	75.00	62.50	100.00	75.00	93.75	+31.25
NOUN:NUM	60.39	50.40	58.09	74.21	47.58	66.74	+8.65
NOUN:POSS	65.00	46.43	60.19	63.04	51.79	60.42	+0.23
ORTH	75.53	55.59	70.47	71.22	59.94	68.63	-1.84
OTHER	40.92	18.95	33.21	38.14	22.49	33.48	+0.27
PART	56.52	44.07	53.50	58.54	40.68	53.81	+0.31
PREP	53.77	34.40	48.33	64.67	41.90	58.33	+10.00
PRON	48.39	33.15	44.31	55.71	43.09	52.63	+8.32
PUNCT	63.53	49.60	60.15	70.07	47.26	63.90	+3.75
SPELL	82.09	58.41	75.94	86.15	53.85	76.92	+0.98
VERB	48.11	20.23	37.71	44.81	21.59	36.88	-0.83
VERB:FORM	64.35	59.15	63.24	71.14	60.85	68.82	+5.58
VERB:INFL	50.00	50.00	50.00	80.00	66.67	76.92	+26.92
VERB:SVA	61.01	68.79	62.42	72.41	74.47	72.82	+10.40
VERB:TENSE	58.10	38.28	52.65	63.50	44.77	58.60	+5.95
WO	51.47	39.77	48.61	64.71	37.50	56.51	+7.9
Total	58.49	38.29	52.91	62.64	40.72	56.55	+3.64

Table C.1: Error type-specific performance of the sentence-level **Baseline** and the document-level **MultiEnc-dec** on BEA-dev. The last column shows the difference in $F_{0.5}$ between document-level and sentence-level systems.

D GEC system output examples

Context	In the chat room, she created a close relationship with eight people.
Source	She talks to them every night, trust them and share her life with them.
Reference	She **talked** to them every night, **trusted** them and **shared** her life with them.
Baseline	She talks to them every night, trusts them and shares her life with them.
Our model	She **talked** to them every night, **trusted** them and **shared** her life with them.

Context	Solar heaters have been introduced in houses instead of water heaters.
Source	Rain water storage system to increase water level.
Reference	**and rain** water storage **systems** to increase water **levels**.
Baseline	Rain water storage system to increase water level.
Our model	**and rain** water storage **systems** to increase water **levels**.

Context	My favourite sport is volleyball. When I am on the beach I like playing with my sister in the sand and then we go in the sea.
Source	It is very funny.
Reference	It is **great fun**.
Baseline	It is very funny.
Our model	It is **great fun**.

Context	It was the first time for me to play basketball.
Source	I think I were very good.
Reference	I think I **was** very good.
Baseline	I think I am very good.
Our model	I think I **was** very good.