Grammatical Error Correction through Round-Trip Machine Translation

Yova Kementchedjhieva
University of Copenhagen
yova@di.ku.dk

Anders Søgaard
University of Copenhagen
soegaard@di.ku.dk

Abstract

Machine Translation operates on the premise of an interlingua which abstracts away from the surface form while preserving the meaning. A decade ago, the idea of using round-trip MT to guide Grammatical Error Correction was proposed as a way to abstract away from potential errors in surface forms (Madnani et al., 2012). At the time, it did not pan out due to the low quality of MT systems of the day. Today much stronger MT systems are available so we re-evaluate this idea across five languages and models of various sizes. We find that for extra large models input augmentation through round-trip MT has little to no effect. For more 'workable' model sizes, however, it yields consistent improvements, sometimes bringing the performance of a *base* or *large* model up to that of a *large* or *xl* model, respectively. The round-trip translation comes at a computational cost though, so one would have to determine whether to opt for a larger model or for input augmentation on a case-by-case basis.

1 Introduction

Grammatical Error Correction (GEC) is the task of detecting and correcting errors in text. It finds application in both assisted writing and second language learning. As training data for the task is scarce, efforts in this space largely focus on transfer learning and data augmentation. In this work, we revisit the use of round-trip Machine Translation in Grammatical Error Correction, as originally proposed in Madnani et al. (2012).

Machine Translation (MT) aims to preserve the meaning of text while mapping its surface form from one language into another. The ideal MT system would be robust to minor perturbations in the input text like a typo or a grammatical error, producing a well-formed translation true to the intended meaning. If the translated text is then backtranslated into the source language, we can expect to see the original content, now free from errors. This

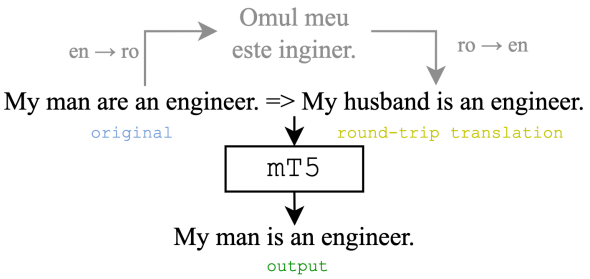

Figure 1: Our approach. The input to the model is a concatenation of the original text and the round-trip translation. Here, English is used as target language and Romanian as pivot language for illustrative purposes; in actual experiments English is the pivot language.

was the premise of the work carried out by Madnani et al. (2012). The statistical phrase-based MT systems of a decade ago, however, were not even close to the ideal, so the authors observed mixed results in their experiments and upon further analysis concluded that the round-trip translation itself introduced too many new errors in the form of both ungrammaticality and loss of meaning. Modern neural network-based MT systems are much stronger than their statistical predecessors. Consider the leap in BLEU score (Papineni et al., 2002) on the widely used WMT2014 English-German data set (Bojar et al., 2014), from 20.7 with phrase-based MT (Wu et al., 2016) to 35.0 with neural MT augmented with noisy backtranslation (Edunov et al., 2018). With a conditional neural language model as a decoder (Schwenk, 2007), modern systems generate highly fluent (i.e. grammatical) outputs.

We explore the impact of this strong MT performance on GEC by augmenting the input to a GEC system through round-trip translation, such that each input sentence is concatenated with a round-trip translation of itself (see Figure 1). We evaluate the effect of this procedure on five languages: German, Russian, Spanish, Czech, and Romanian (DE, RU, ES, CS, RO). In our experiments, we fine-tune the multilingual pre-trained

Findings of the Association for Computational Linguistics: EACL 2023, pages 2208–2215
May 2-6, 2023

language model mT5 (Xue et al., 2021), which is available in a range of sizes. The XXL variant is currently the state-of-the-art on DE, RU, and CS (Rothe et al., 2021). However, mT5-XXL, with its 13B parameters, is out-of-scope for most academic research and impractical for deployment in many application. Therefore, we experiment only with the three smaller variants, BASE, LARGE and XL.

We find that round-trip translation successfully guides the correction of grammatical errors in BASE models for all languages, with improvements of up to 4.1 points on the F-score (for RU). For LARGE models, it still benefits three out of five languages, leaving scores on the other two unchanged. For XL models, it has a negligible effect in either direction, showing that these models are sufficiently strong by themselves and subsume the knowledge an MT model can provide. For some BASE and LARGE configurations, the round-trip translation augmentation closes the gap between a model of a given size, e.g. LARGE-RO, and its larger counterpart, e.g. XL-RO. Since round-trip translation has an added computational cost itself, one would have to weight the costs and benefits on a per-case base to determine whether a larger model with bare input or a smaller model with augmented input is more suitable for a given application.

2 Background

Machine Translation makes various appearances across research in Grammatical Error Correction. Modeling approaches and training tricks originally developed in the context of MT have been successfully adapted to GEC (Yuan and Felice, 2013; Junczys-Dowmunt et al., 2018; Rozovskaya and Roth, 2016; Yuan and Briscoe, 2016). The concept of backtranslation has been used to generate synthetic data for GEC (Kiyono et al., 2019; Koyama et al., 2021). In all of these works, MT research provides the methods but there is no actual cross-lingual translation happening. The 'translation' in this case is from ungrammatical text to grammatical text in the same language. Zhou et al. (2020) perform actual translation of Chinese text into English using MT systems of varying quality as a way to generate ungrammatical English data, which they then pair with gold standard targets to obtain a synthetic training corpus. In contrast to such works, our work explores the potential of round-trip translation as an intermediate step in the process of GEC, active both during fine-tuning and inference. The goal here is to make use of the knowledge one can extract from parallel MT data, generally much more abundant than GEC data.

Most similar to our work is that of Madnani et al. (2012), who perform round-trip translation of an input in eight pivot languages with Google Translate and use a lattice to combine all hypotheses into a final output. The motivation behind using multiple pivot languages is to ensure meaning preservation on one hand, and to increase the chance of all errors being corrected on the other. The authors observe some successes but also numerous failures in the predictions of their model, attributing the latter to new errors of disfluency and loss of meaning introduced by Google Translate, which at the time was based on statistical MT. In the decade since that work was published, MT has undergone a paradigm shift from statistical to neural network-based methods, marked by large improvements in performance (Edunov et al., 2018). It is therefore time to revisit the potential gains from round-trip MT for GEC.

As we recognize that GEC aims for minimal and necessary revisions of the input whereas round-trip translation can result in valid but unnecessary lexical and syntactic changes, we condition the generation of the final output on both the input sentence and the round-trip translation, in an approach akin to multi-source automatic post-editing (Knight and Chander, 1994; Chatterjee et al., 2015).

3 Method

In general terms, our approach is one of sequence-to-sequence text generation with input augmentation: for a given input sentence, we obtain a round-trip translation and feed a string concatenation of the original sentence and the round-trip translation, separated by the symbol sequence '=>', to a sequence-to-sequence model.

3.1 Model

Recently, Rothe et al. (2021) set a new state-of-the-art in GEC using an XXL-sized mT5 model. mT5 is a multilingual seq-to-seq bitransformer model, pre-trained on 101 languages (Xue et al., 2021). They pre-trained a single model on a vast amount of synthetic GEC data for four languages, English, Czech, Russian and German, and fine-tuned individual models for each language. They showed that a BASE-sized model often lagged behind earlier state-of-the-art results, whereas an XXL-sized model outperformed them often with a consider-

Lang	Data	Size
DE	Falko-Merlin (Boyd et al., 2014)	19K
RU	RULEC-GEC (Rozovskaya and Roth, 2019)	5K
ES	COWS-L2H (Davidson et al., 2020)	10K
RO	RoGEC (Cotet et al., 2020)	7K
CS	AKCES-GEC (Náplava and Straka, 2019)	42K

Table 1: Datasets used for finetuning and their train size.

able margin.[1] Due to computational constraints, we carry out experiments with model sizes up to and including XL.

3.2 Data

We use data in five languages: DE, RU, ES, CS and RO.[2] We carry out continued pre-training of mT5 for GEC on real data where available, and on synthetic data otherwise. For DE and RU we use cLang-8 data (Rothe et al., 2021). For ES, we use Lang-8 (Koyama et al., 2020), which we manually clean up (see more details in Appendix A). For RO we sample 100k sentences from the synthetic dataset of Cotet et al. (2020) and for CS we generate 100k sentences using the method of Náplava and Straka (2019) based on text from the WMT News Crawl (Barrault et al., 2019). We randomly split all data 90:10 for training and validation. Continued pre-training is done with the same objective as used for fine-tuning—we feed ungrammatical text (optionally concatenated with a round-trip translation) and predict grammatical text. Following this step, we do fine-tuning on the datasets listed in Table 1.

All data that does not come pre-tokenized is tokenized using spaCy (Honnibal and Montani, 2017) except CS, which is not covered by spaCy so for this language we use Stanza (Qi et al., 2020).

3.3 Round-trip translation

In contrast to Madnani et al. (2012), we stick to a single round-trip translation, recognizing the computational cost of this added step. We experiment with English as a fixed pivot language for all target languages. We translate pre-training data using models available in the HuggingFace library (Wolf et al., 2020), chosen for their strong performance: for RU and DE we use `facebook/wmt19` models, and for the rest we use `Helsinki-NLP/opus-mt`. For fine-tuning data we use Google Translate[3], assuming that it is the best translator available.

Training details can be found in Appendix B.

4 Results

The main results of our work are reported in Table 2. We report precision (P), recall (R) and F0.5 score (F), as measured using the M^2 package (Dahlmeier and Ng, 2012). We see that guidance from round-trip translation leads to consistent improvements for models based on mT5-BASE, most notably improving the F-score for RU by 4.1 points. Among LARGE models, consistent performance improvements are observed for RU and CS, for RO the performance gain is reduced but still considerable, whereas for DE and ES the input augmentation has no effect at all (so we do not consider these two languages in experiments with an XL model). Among the three XL models, variable results are observed with either a small increase or a small decrease in performance (of 0.5 points at most). From these observation, we can conclude that the round-trip translation benefits smaller models, whereas larger ones subsume the knowledge this input augmentation technique provides.

The blue boxes in the table mark cases where the round-trip translation brings the performance of a smaller model up to or above that of a larger one. In these cases, one has the choice to use a larger model without input augmentation or a smaller one with input augmentation. The factors that would determine this choice are compute availability, access to cloud platforms, and speed requirements, among others. If one has limited GPU memory to work with, but has access to a high-quality translation cloud service, the choice of a smaller model with input augmentation may be more appropriate.

4.1 Round-trip translation

Although round-trip translation is expected to correct errors while preserving meaning, we cannot rely on it alone as a method for grammatical error correction, due to potential lexical and syntactic substitutions. This becomes apparent when we treat the output of the round-trip translation as GEC predictions and evaluate them against the gold-standard targets. The results, shown in the last row of Table 2 (MT), are considerably lower

[1] Model sizes in between were not explored.

[2] See App. C for other languages we considered.

[3] https://cloud.google.com/translate; We were able to carry out all translation at no cost, taking advantage of a promotion available at the time of writing, wherein new users get $300 in free credits.

	DE			RU			ES			CS			RO		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
BASE	74.9	58.0	70.8	59.5	15.5	38.0	57.9	35.8	51.5	78.9	60.5	74.4	68.9	46.5	62.9
BASE + MT	76.0	61.5	72.6	60.9	18.8	42.1	58.5	39.4	53.4	79.4	65.0	76.0	70.1	55.0	66.5
L	77.	62.7	73.6	60.4	22.8	45.4	61.5	39.1	55.2	80.9	65.6	77.3	72.2	50.7	66.5
L + MT	76.4	64.3	73.6	63.6	25.8	49.2	60.4	41.0	55.2	81.9	70.5	79.3	71.7	58.3	68.6
XL				64.5	25.6	49.5				81.7	69.9	79.0	72.3	56.9	68.6
XL + MT				61.8	28.1	49.9				82.0	70.8	79.5	70.3	60.5	68.1
SOTA	-	-	76.	-	-	51.6	-	-	57.3	-	-	83. 2	-	-	53.8
MT	38.9	50.9	40.8	20.6	48.1	23.3	27.7	39.1	29.4	19.8	33.2	21.5	40.9	51.4	42.7
BASE+	68.7	57.3	66.1	45.0	19.0	35.3	51.2	37.8	47.8	71.8	60.5	69.2	59.2	49.2	56.9

Table 2: Main results. SOTA refers to results from Rothe et al. (2021) for DE, RU and CS, results from Flachs et al. (2021) for ES and Cotet et al. (2020) for RO. Experiments with XL models were not performed for DE and ES since for these languages even in the LARGE configuration, the round-trip translation does not help. Blue boxes mark instances where an augmented smaller model performs comparably to a larger model.

	DE				RU			
	F-MT	P	R	F	F-MT	P	R	F
BASE	-	74.9	58.0	70.8	-	59.5	15.5	38.0
GT	40.8	76.0	61.5	72.6	23.3	60.9	18.8.	42.1
FB	35.6	74.7	62.8	72.0	17.9	58.5	16.9	39.2

Table 3: Comparison of MT systems. GT: Google Translate, FB: `facebook/wmt19`. F-MT refers to the F-score of the round-trip translation as prediction.

than the full system results in upper rows, even in comparison to the BASE setting.

4.2 Alternative MT systems

To determine the importance of a high-quality MT system for the success of our method, we carry out experiments with an alternative translation system, `facebook/wmt19`, used to obtain round-trip translations for the fine-tuning data in RU and DE. The results from training a BASE-size model on this data are shown in Table 3 alongside the main results with this model size. Although `facebook/wmt19` scores substantially lower than Google Translate when the round-trip translation alone is compared to the gold standard (F-MT), clear gains from using the round-trip translations for input augmentation can be observed.

4.3 Input augmentation v. Data augmentation

To determine the role of input augmentation as compared to the more common method of data augmentation, we train BASE models with the round-trip translations as additional data, i.e. we extend the training set with the pairings of round-trip translated sentences and their gold-standard targets, thus doubling its size. As can be seen in the last row of Table 2 (BASE+), this leads to worse performance, likely because the revisions from round-trip translated sentences to gold-standard ones do not only contain grammatical error corrections, but also some 'unnecessary' (from the perspective of GEC) lexical and syntactic changes.

4.4 Overall performance

The results we obtain fall short of the state-of-the-art on four out of five languages. For DE, RU and CS this is no surprise considering the size of the model used by Rothe et al. (2021), which renders their achieved improvements irrelevant in most practical contexts. We did not experiment with the data augmentation strategy used in Flachs et al. (2021)—this would have likely lead to a higher baseline performance in our setup as well. For RO, on the other hand, we see a large improvement over the work of Cotet et al. (2020) even with a BASE model, and an almost 15 point improvement overall.

5 Conclusion

The goal of this study was to measure the benefits of round-trip machine translation for the task of grammatical error correction. Transferring knowledge from an MT model to a GEC model through input augmentation proved effective for smaller models, sometimes bringing their performance up to that of their larger counterparts. In this work, we chose English as a pivot language due the abundance of MT work on this language. Future work could explore alternative pivot languages, option-

ally ones that are related to the language of the GEC data, as this may results in higher lexical and syntactic consistency between inputs and round-trip translations and thus better guidance for the correction of grammatical errors.

6 Limitations

The computational cost of the method proposed here cannot be measured in a universal sense, since (a) we have no way of determining the computational requirements for a call to the Google Translate API, (b) while one could run translation locally, given a good enough translation system, the exact computational costs of that process would also depend on the size of the local translation model, with trends in MT also shifting towards models of growing size. It is therefore only on a case-by-case basis that one can determine whether in their specific case it is more efficient to perform GEC with a larger model or to use a smaller model in combination with performing round-trip MT.

7 Acknowledgements

This work was funded by Innovations Fund Denmark under the AutoAI4CS and PIN projects.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the planet of the APEs: a comparative study of state-of-the-art methods for MT automatic post-editing. 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)*, pages 156–161, Beijing, China. Association for Computational Linguistics.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631. IEEE.

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.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

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.

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.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*.

Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. Comparison of grammatical error correction using back-translation models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135, Online. Association for Computational Linguistics.

Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 204–211, Marseille, France. European Language Resources Association.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Exploring grammatical error correction with not-so-crummy machine translation. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada. Association for Computational Linguistics.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47, Doha, Qatar. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

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.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Oleksiy Syvokon and Olena Nahorna. 2021. Ua-gec: Grammatical error correction and fluency corpus for the ukrainian language.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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

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

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. Improving grammatical error correction with machine translation pairs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online. Association for Computational Linguistics.

A Spanish data for continued pre-training

Lang8 data can be noisy, due to people adding meta-comments to the text they post (often in their native language) or the edits they propose. Many of these instances can be detected based on length mismatch or foreign scripts. So we remove any data points where the number of space-separated tokens on one side mismatches the other by more than three and any lines that contain non-Latin characters. This leaves us with 182,039 data points for continued pretraining.

B Training

We use identical training settings for BASE and LARGE models. In bare-input experiments (original input only) we set the maximum input length to 256 and in experiments with augmented input, to 512. The maximum output length is always 256. For continued pre-training, we use a learning rate of 0.001, following Rothe et al. (2021). For fine-tuning, we experiment with 0.001, 0.0005, and 0.0001, choosing the best one per language based on the validation loss in BASE experiments and reusing it for LARGE experiments.[4] For XL models, we halve the input and output lengths due to computational constraints and we halve the learning rates as we observed that the learning rates used for smaller models result in quick overfitting. We follow Rothe et al. (2021) in setting the batch size to 1,048,576 tokens per batch, which for bare-input experiments amounts to an effective batch size of 2048 and for experiments with augmented input, to 1365.[5] In all experiments, we use the Adafactor optimizer (Shazeer and Stern, 2018) and train until the validation loss stops improving.

[4]We note that learning rate has a considerable impact on the results of up to 5 F0.5 points for different configurations.

	P	R	F
SOTA	73.3	63.2	71.1 *
BASE	81.4	64.7	77.4
LARGE	81.6	71.6	79.4

Table 4: Baseline results for Arabic. * computed by us from the global recall and precision scores, as the authors report F1 rather than F0.5

C Other languages

Arabic In the course of this work, we considered experimenting with the QLAB dataset (Mohit et al., 2014) for grammatical error correction in Arabic. We later determined that the cost of the round-trip translation of this data exceeds our resources: due to the non-UTF script used by Arabic, the 19,411 training data points in QLAB amount to almost 10M characters and Google Cloud API charges by the character. Since we did train baseline models on this data, however, we report the results here (see Table 4), for future reference.

Data for continued pretraining in the amount of 100k sentences was generated with the method of Rothe et al. (2021) as applied to a sample of 100k sentences again from the WMT News Crawl. The data was tokenized using NLTK (Bird et al., 2009).

Ukrainian We considered experimenting with the newly introduced Ukrainian dataset UA-GEC (Syvokon and Nahorna, 2021) as well but faced challenges in the segmentation of the data—in contains entire documents, often longer than the maximum sequence length of standard transformer-based models. We considered splitting those into paragraphs on new line symbols, but that produced

[5]These batch sizes are achieved with gradient accumulation, with an actual batch size of 4 for BASE models, 2 for LARGE models and 1 for XL models. We train the former two on RTX GPU cards (24 GB) and the latter on A100 (40 GB).

many nonsense data points such as section headings and some stand-alone meta-text strings.