Grammatical Error Generation Based on Translated Fragments

Eetu Sjöblom and **Mathias Creutz** and **Teemu Vahtola**
Department of Digital Humanities, Faculty of Arts, University of Helsinki, Finland
`{eetu.sjoblom,mathias.creutz,teemu.vahtola}@helsinki.fi`

Abstract

We perform neural machine translation of sentence fragments in order to create large amounts of training data for English grammatical error correction. Our method aims at simulating mistakes made by second language learners, and produces a wider range of non-native style language in comparison to state-of-the-art synthetic data creation methods. In addition to purely grammatical errors, our approach generates other types of errors, such as lexical errors. We perform grammatical error correction experiments using neural sequence-to-sequence models, and carry out quantitative and qualitative evaluation. A model trained on data created using our proposed method is shown to outperform a baseline model on test data with a high proportion of errors.

1 Introduction

Grammatical error correction (GEC) is the task of detecting and correcting grammatical errors in texts, typically written by second language learners. Current state-of-the-art GEC approaches are based on neural machine translation (NMT) (Grundkiewicz et al., 2019). As in other natural language processing tasks, neural approaches to GEC rely on large quantities of task-specific data, that is, sentence pairs consisting of erroneous source text coupled with corrected target text. However, in-domain GEC data is scarce, and a number of solutions to the data sparsity problem have been proposed recently, often by introducing artificially created GEC data into the training process.

Some error generation approaches also depend on error-annotated authentic learner data. For example, Felice and Yuan (2014) introduce errors probabilistically with error probabilities that are estimated using a learner corpus. Rozovskaya et al. (2014) train error detection and classification models on annotated data, focusing on verb errors. Other methods dispense with the need for annotated data, such as approaches based on inverted spell-checkers and heuristic error generation (Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019).

To alleviate the data sparsity problem, in this work we propose to use NMT to produce artificial training data, simulating real errors made by language learners. For instance, to produce English text with errors, we use NMT models to translate sentence fragments from other languages to English, and then combine the translated fragments to form our erroneous source data. Similar machine translation approaches to GEC data creation have been proposed before. For example, Rei et al. (2017) use a statistical machine translation model trained on reversed learner data, using the corrected sentences as source data and erroneous sentences as targets. Kasewa et al. (2018) extend this approach and use an NMT model to produce errors. Htut and Tetreault (2019) perform extensive experiments on several neural models, likewise trained on learner data to generate errors.

Our contribution is to split the foreign-language source sentences into shorter fragments in order to limit the context that is available to the machine translation system. The rationale for doing this is to produce text that contains artefacts from the foreign language. Since the NMT system needs to translate shorter fragments without the proper context, we expect it to produce more literal translations and to be less able to produce correct agreement between different parts of speech. Additionally, polysemy may prompt the system to suggest translations of a synonym in the foreign language, which is not a synonym in English. The creation of synthetic training data involves further steps, which are described in Section 2. Model training is explained in Section 3. In Section 4 we evaluate our

approach quantitatively against a strong baseline (Grundkiewicz et al., 2019) and make some qualitative assessments.

2 Training data

The creation of our training data involves the following steps:

1. English sentences aligned with sentences of other languages are used as data.[1] Our parallel text data are retrieved from the OpenSubtitles (Lison and Tiedemann, 2016) and Europarl (Koehn, 2005; Tiedemann, 2012) collections.[2]

2. The non-English sentences are split randomly into chunks of an average length of three word tokens.

3. Each sentence chunk in isolation is translated into English using OPUS-MT machine translation models from HuggingFace (Tiedemann and Thottingal, 2020). N-best lists containing up to ten alternate translations for each chunk are produced.

4. Full English sentences are created by concatenating chunks from the n-best lists. Ten different alternate full sentence translations are generated for each source sentence by combining chunks at random, proportionally to the translation scores of the chunks. Our aim is to obtain English translations that contain errors influenced by the source language. The original English sentence from the parallel corpus serves as a correct reference translation. Examples are shown in Table 1.

5. In theory, for each sentence in our data we now have ten artificially created, erroneous English sentences. However, many of the synthetic sentences do not resemble authentic human-produced erroneous sentences. We therefore discard a significant portion (60 %) of the synthesized sentences by sampling for an error distribution that is closer to the error distribution of authentic data, represented by our development sets. This leaves us with just

23 % of the words of our original set, reflecting the fact that longer sentences are more likely to be discarded.

6. The above mentioned sampling of sentences requires us to be able to compare error distributions between authentic and synthetic data. First we POS tag the sentence pairs and align them automatically using minimum string edit distance coupled with some heuristics taking into account part of speech and inflection. The alignment algorithm is similar but not identical to ERRANT (Bryant et al., 2017; Felice et al., 2016). This procedure is illustrated in Table 2. From the alignments we extract trigrams consisting of a correction operation in the context of one preceding and following token, such as *PRP ins(MD) VBP* ("insert a modal verb between a pronoun and a verb in non-third person singular form") or *ins(DT) ins(JJ) NN* ("insert an adjective between an inserted determiner and a noun in singular"). These automatically extracted trigrams constitute our correction types. Their frequency distributions are not the same across the authentic and synthetic data. We filter the synthetic data by keeping sentence pairs that contain combinations of correction types that are highly likely to occur in authentic data and discard sentence pairs with low-probability correction types.

2.1 Final training sets

We carry out experiments using systems trained on four different training sets. We create one data set using our method that matches the word count of the Baseline comparison. In addition, we create two smaller data sets using both the Baseline method and ours on the same correct target sentences in order to control the effects of data domain.

- **Baseline:** We compare our own training scheme to a system trained on the training set created by Grundkiewicz et al. (2019). They propose an unsupervised data generation method based on confusion sets from spellcheckers. For each source sentence in the news crawl data used for training, they replace a random number of tokens with a substitute from the vocabulary item's confusion set. In

---

[1]These languages, which have been chosen to represent both European and Asian languages from different language families are the following: Danish, Dutch, Finnish, French, German, Italian, Japanese, Korean, Latvian, Portuguese, Russian, Spanish, and Swedish.

[2]Available for download at: `https://opus.nlpl.eu/`

de-src	Während / du / bewusstlos im / Krankenhaus / lagst, sind / die Ärzte mit / diesen / Testergebnissen / zu / Daniel gekommen.
en-tgt	**During / you / unconscious in / Hospital / the / doctors with / the / Test results / to / Daniel came.**
en-ref	While you were unconscious in the hospital, the doctors came to Daniel with these test results.
ru-src	И никогда не / переставал / думать о / тебе.
en-tgt	**And never / I stopped / to think about / You.**
en-ref	I never stopped thinking about you.
fr-src	Il est / vrai que toutes les / histoires ne peuvent avoir une fin heureuse, mais pour Jules / Daly, la rêveuse de Buffalo, / l'histoire ne / fait que commencer.
en-tgt	**It is / true that all / stories can't have a happy ending, but for Jules / Daly, Buffalo's dreamer, / history / Just start.**
en-ref	It is true not all tales have happy endings, but then for Jules Daly, the dreamer from Buffalo, the story is just beginning.
ja-src	もし君が生き残れたら / 一生懸命に働いたからだ
en-tgt	**And if you survive, / Because you worked hard.**
en-ref	If you live, you have worked very hard indeed.
fi-src	Koulu / on - / lähettänyt / minut useammalle / terapeutille kuin / sinulla / on ollut huonoja / treffejä.
en-tgt	**School / is / sent / me to more / for a therapist / you / has been bad / Date.**
en-ref	This school has sent me to more therapists than you've had bad dates.

Table 1: Sentences in other languages (*-src) are split into chunks (e.g., / *bewusstlos im* /), and each chunk in isolation is translated automatically into English. By concatenating the chunks we obtain English sentences containing errors (en-tgt), for which correction hypotheses exist in the form of the English reference translations (en-ref).

addition, they probabilistically delete and insert random tokens, as well as swap adjacent tokens in the sentence. They also introduce additional noise at the character level using similar operations. Although these operations introduce some syntactic and word order mistakes, the method does not excel at producing more complex syntactic errors, errors that require extensive reordering of the sentence, or errors that result from L1 influence.

- **Chunks:** We produce a training set using our method, which is sampled to contain the same number of words as the Baseline (4.6 billion words).

- **Chunks-small:** We produce a training set using our method such that the data set contains only unique target sentences. This smaller set contains approximately 650 million word tokens and allows for faster model training.

- **Baseline-small:** We use the Baseline data creation method on the same target sentences as in the Chunks-small set.

3 Model training

We build on the system described in Grundkiewicz et al. (2019). We choose not to make changes to the model or training parameters in order to isolate the effects of our data creation method and ensure a fair comparison. The same training setup is used for all models, with modifications only in the training sets. Specifically, we use their "Transformer Big" architecture, with 6 self-attention layers, 16 attention heads, embeddings vectors of size 1024, and feed-forward hidden size of 4096 with ReLU activation functions. We also tie the encoder, decoder, and output embeddings.

We also adopt the training setup of Grundkiewicz et al. (2019), and train our models with the Marian toolkit (Junczys-Dowmunt et al., 2018). The models are first pretrained on the synthetic data for a maximum of 5 epochs. After pretraining, we finetune the best model checkpoint using the following corpora: FCE (Yannakoudakis et al., 2011), NUCLE (Dahlmeier et al., 2013), W&I-LOCNESS (Bryant et al., 2019; Granger, 1998), and Lang-8 (Mizumoto et al., 2012). We use the W&I-LOCNESS development set for validation during training.

We use early stopping with a patience of 10 with ERRANT $F_{0.5}$ score on the W&I+locness development set used as the early stopping criterion. The checkpoint with the highest $F_{0.5}$ score is chosen for further finetuning. We choose the ADAM optimizer, a learning rate of 0.0002 and a linear warmup for 8k updates. We use Marian's option to dynamically fit mini-batches to GPU memory, and train our models using 4 Nvidia Volta V100 GPUs (32GB RAM). In addition, we use strong regularization, which has been found useful in GEC systems, with dropout probabilities of 0.3 between

	W&I-LOCNESS	YKI
Baseline	**66.44**	52.63
Chunks	65.44	**53.41**
Baseline-small	*60.09*	46.66
Chunks-small	59.89	*49.73*

Table 3: $F_{0.5}$ scores for the four models on the two test sets. $F_{0.5}$ is a weighted harmonic mean of precision and recall, where precision is accentuated.

Learner sentence:	We had enjoy time .
Correction:	We had a great time .
Alignment:	PRP VBD del(VB) ins(DT) ins(JJ) NN .
Synthetic sentence:	You be the the old donkey of the forestry
Correct reference:	You 'll be the oldest donkey in the forest .
Alignment:	PRP ins(MD) VBP del(DT) DT inf(JJS) NN del(IN) ins(IN) DT typ(NN) ins(.)

Table 2: Pairs of sentences with alignments. The upper example is an authentic sentence produced by a language learner accompanied by a correction (target hypothesis) proposed by a teacher. The alignment is a sequence describing how to modify the learner sentence into the corrected one. It reads as follows: *PRP*: keep pronoun ("we"), *VBD*: keep verb in past tense ("had"), *del(VB)*: delete verb in infinitive ("enjoy"), *ins(DT)*: insert determiner ("a"), *ins(JJ)*: insert adjective ("great"), *NN*: keep noun in singular ("time"), *.*: keep punctuation. The lower example is analogous, but the alignment is between a synthetically produced sentence and the correct English reference. This alignment sequence contains a few more correction types: *inf(JJS)*: change inflection of adjective into superlative ("oldest"), *typ(NN)*: fix typo in noun in singular (the word "forestry" is here classified as a spelling error by the algorithm).

layers, 0.1 for self-attention and feed-forward layers, 0.3 for entire source token embeddings, and 0.1 for target embeddings.

4   Evaluation

We report results on two different data sets. We use the W&I-LOCNESS set, which was used as official test data in the BEA19 GEC shared task (Bryant et al., 2019), as well as a subset of the English portion of learner texts derived from the Finnish National Certificates of Language Proficiency exams (YKI).[3] We do not use the YKI data as a blind test set, but instead use it to qualitatively analyze differences in model predictions. Still, no part of the YKI data was used during training or development of the models.

We compare our results with those reported by Grundkiewicz et al. (2019), whose system achieved first place in the BEA19 GEC shared task. However due to limited resources we do not train an ensemble of models, but instead take a single left-to-right

model from Grundkiewicz et al. (2019) as baseline. Their best system uses an ensemble approach with right-to-left and language model reranking and achieves a higher $F_{0.5}$ score of 69.47 on the W&I-LOCNESS test set.

The upper part of Table 3 compares our Chunks model with the baseline by Grundkiewicz et al. (2019). The Baseline model performs best on W&I-LOCNESS with a one absolute point difference compared to our model. However, our model outpeforms the Baseline on YKI. These results suggest that our data creation method might be suitable when correcting noisier source sentences, as YKI generally contains more challenging language with more errors than W&I-LOCNESS.

The lower part of Table 3 demonstrates that the trends are the same for the smaller models, in which we match the data domain in training. That is, the Baseline is no longer trained on news data but on OpenSubtitles and Europarl. The results are lower overall due to the smaller data size. The Baseline outperforms our model on W&I-LOCNESS also in this setting, although by a smaller margin. However, the performance gap on YKI increases by approximately two absolute points in favor of our model, offering additional support that our method can improve performance on noisy data. To better understand differences between the models, we examine their predictions on the same source sentences, as described in the next section.

4.1   Qualitative assessment

We have taken a closer look at the corrections made by the Baseline and Chunks models on the 320

---

[3]Available for research purposes from the Centre for Applied Language Studies at the University of Jyväskyla, Finland: http://yki-korpus.jyu.fi/

sentences in the YKI data set. The results are surprisingly similar although the models have been trained on different text corpora, into which errors have been introduced using different methods. The Chunks model suggests slightly more corrections on average than the Baseline, yielding a somewhat higher recall and lower precision.

It is interesting to see that the Chunks model performs quite well on misspelled words (*broblems*, *i'ts*, *beatyfull*) although it has not been explicitly trained to correct spelling mistakes, in contrast to the Baseline method. In the training data of the Baseline, spelling errors have been introduced by random sampling, whereas the models based on machine translated data generally do not contain any spelling mistakes, as machine translation does not generate them. Yet, it appears that the Chunks model corrects spelling errors at least as well, if not better, than the Baseline. The latter model leaves word forms, such as *wery*, *nicier* and *higing* (for *hiking*) unchanged.

When it comes to choosing the correct spelling in context, the Chunks model distinguishes between the different usages of *prize* and *price* ("*The prize for you is betveen 1500-1700 euros.*"), and it has in fact been trained on almost 4000 sentence pairs in which *prize* is corrected to *price* in context. The Baseline does not make this correction.

None of the models manage to correct the sentence "*I have old but wery fine cun selling.*". Firstly, the models fail to change *cun* into *gun*. Secondly, one could have expected the Chunks model to see the connection between *selling* and *for sale*, since there are 800 training examples containing that substitution, but for some reason this particular test sentence does not trigger the desired change.

Many of the sentences in the YKI corpus are indeed hard to interpret without broader world knowledge; The Chunks model corrects the sentence "*If it isn't help then you will ask for help to polishman*" into "*If it doesn't help then you will ask the Polishman for help.*". However, the correct person to ask for help here would be the police man. In another sentence, "*I begin hobbies about 12 yers.*", the model would somehow need to understand that the person picked up hobbies at the age of twelve rather than twelve years ago.

Additionally, we have examined the W&I-LOCNESS development set, although it has been used as a stopping criterion in the training, which may bias the results slightly. The $F_{0.5}$ scores on the dev set are the same for both the Baseline and the Chunks model (52.6 %). This is considerably lower than for the final test set (65.4 - 66.4 %), suggesting that the test set is less challenging than the dev set. Compared to the YKI data, even the W&I-LOCNESS dev set seems cleaner and appears to contain fewer mistakes. It is hard to see significant differences in performance between the models. For 65 % of the sentences, the Baseline and the Chunks model produce exactly the same corrections. The corresponding figure for the YKI set is 57 %.

5   Discussion and conclusion

We have shown that our model rivals a competitive baseline, a left-to-right model by Grundkiewicz et al. (2019), which was one component of an ensemble model that performed best in the BEA19 GEC shared task. We did not yet train our own ensemble model, but we expect to see similar improvements in performance in future experiments.

Our results show that two models can perform on par, although they have been pretrained on different training corpora and using different error simulation techniques. In addition, the Chunks model outperforms the Baseline in noisy conditions. In the future, we would like to analyze further techniques for modeling challenging types of errors, which originate from structures that differ between the target language and the native languages of the language learners.

6   Acknowledgments

References

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. https://doi.org/10.18653/v1/W19-4406 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.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. https://www.aclweb.org/anthology/W13-1703 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, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in esl sentences using linguistically enhanced alignments.

Mariano Felice and Zheng Yuan. 2014. https://doi.org/10.3115/v1/E14-3013 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 *Learner English on Computer*, pages 3–18. Addison Wesley Longman.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. https://doi.org/10.18653/v1/D19-5546 Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. https://doi.org/10.18653/v1/W19-4427 Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Phu Mon Htut and Joel Tetreault. 2019. https://doi.org/10.18653/v1/W19-4449 The unbearable weight of generating artificial errors for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 478–483, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. https://doi.org/10.18653/v1/W18-2716 Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. https://doi.org/10.18653/v1/D18-1541 Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT Summit 2005*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. https://www.aclweb.org/anthology/C12-2084 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.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. https://doi.org/10.18653/v1/W17-5032 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.

Alla Rozovskaya, Dan Roth, and Vivek Srikumar. 2014. https://doi.org/10.3115/v1/E14-1038 Correcting grammatical verb errors. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–367, Gothenburg, Sweden. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenc of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. https://www.aclweb.org/anthology/P11-1019 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.