Improving Precision of Grammatical Error Correction with Cheat Sheet

Anonymous ACL submission

Abstract

In this paper, we explore two approaches of generating error-focused phrases and examine whether these phrases can lead to better performance in grammatical error correction for the restricted track of BEA 2019 Shared Task on GEC. Our results show that phrases directly extracted from GEC corpora outperform phrases from statistical machine translation phrase table by a large margin. Appending error+context phrases to the original GEC corpora yields comparably high precision. We also explore the generation of artificial syntactic error sentences using error+context phrases for the unrestricted track. The additional training data greatly facilitates syntactic error correction (e.g., verb form) and contributes to better overall performance.

1 Introduction

Grammatical Error Correction (GEC) is a natural language processing (NLP) task of automatically detecting and correcting grammatical errors in the text. With the ever-growing number of second language learners of English and demand to facilitate their learning with timely feedback, GEC has become increasingly popular and attracted much attention in both academia and industry in recent years. In a typical GEC task, for example, *Travel* and *bored*. in the sentence ***Travel** by bus is expensive and **bored*** needs to be first detected as incorrect and then be modified to their correct forms (*Travelling* and *boring*). Various approaches have been proposed to solve this problem including language modeling, rule-based classifier, machine-learning based classifier, machine translation (MT), and etc. (Ng et al., 2013, 2014). In the past few years, both GEC-tuned statistical machine translation (SMT) and neural machine translation (NMT) using sequence-to-sequence (seq2seq) learning have demonstrated to be more effective in grammatical error correction than other approaches (Chollampatt and Ng, 2017, 2018; Ge et al., 2018; Zhao et al., 2019).

Just as other machine translation tasks, the quantity and quality of data play an important role in the MT approach to grammatical error correction. While several recent studies have focused on generating artificial grammatical error sentences (e.g. Rei et al., 2017; Kasewa et al., 2018), the current study explores how error-focused phrases influence the performance of grammatical error correction. There are slightly over half million error-contained sentences in the training data provided by the BEA 2019 Shared Task, and the total number of errors is over 1.3 million, which means there are on average 2 or 3 errors in each error sentence. Our intuition is that multiple errors in one sentence can be challenging for MT models to learn and generalize, especially when the amount of training data is limited. Thus, by augmenting the training data with error-focused phrases, which we term as "cheat sheet", MT models can directly "see" the errors and their corrections. We predict that this will lead to better overall performance and precision in particular. We examine two ways of creating cheat sheet–one extracting error and surrounding context and the other one extracting from SMT phrase table (§2). Phrases extracted from the first method is also used to generate artificial **syntactic** error sentences for the unrestricted track of the shared task (§3). We run both SMT by Moses (Koehn et al., 2007) and multi-layer CNN seq2seq NMT (Chollampatt and Ng, 2018) for our training data in restricted (original training + cheat sheet) and unrestricted (original training + cheat sheet + syntactic pseudo corpus) settings (§4). In general, our results show that cheat sheet created with error and surrounding context does lead to an improvement in precision. However, compared to current state-of-the-art results, the recall of our models is considerably low. These results and future work are discussed in the last section.

2 Cheat Sheet

2.1 Error+Context Dictionary

Artificial Error Generation has been a long-studied technique for creating more training data for Grammatical Error Correction systems. In the previous study, there are two types of methods of generating an artificial error, one making use of the real learner data statistics and the other treats all types of errors uniformly. Through experiments, it has been shown that accuracy improves when training and test data are more similar to each other (Felice, 2016). This observation is one reason that motivates us to use directly the extracted parallel phrase dictionary from the combined `m2` files as part of our training data since the dictionary preserves the original error distribution. In the dictionary, each pair of the phrases contains one edit in the `m2` file and contains one context word on both sides of the edit (one context word if the edit is at the start of the end of the sentence). The instances in the dictionary have shorter lengths compared to the parallel sentences.

2.2 SMT Phrase Translation Table

Generating a large table of phrase pairs is an integral part of statistical machine translation. These phrase pairs and their corresponding scores (e.g., translation probability and lexical weighting) are the knowledge source during translation/decoding. These phrases are not linguistically well-formed (e.g., noun phrases and prepositional phrases). Rather, they are just sequences of words of arbitrary length. One major difference between the error+context approach and this one is that error is always centered in the former approach, while an error can appear in any position in a phrase in this one.

The Moses SMT system (Koehn et al., 2007) was used to generate a phrase translation table. We used Giza++ (Och and Ney, 2003) for word alignment, and a 3-gram language model trained on 2 million sentences from the AFP news corpus[1] with KenLM (Heafield, 2011). Our input and output sentence length was limited to 40 to ensure the quality of the phrase table, as longer sentences are harder to train using SMT because of their complex syntax and long dependency structures (Bach, 2012). We then extracted phrase pairs with five or more words and the direct translation probability over 95%. Phrase pairs that were same on the error and correct side were also discarded.

3 Pseudo Corpus with Syntactic Errors

We can define syntactic errors as errors that are grammatically incorrect but in most cases, the meaning is still conveyed as compared to semantic errors where the learner fails to convey the desired meaning across to the reader but the sentence structure is correct. Observing syntactic errors is crucial in improving grammatical error correction since they have a direct correlation to grammatical errors since syntactic error produce grammatically incorrect sentences.

Learners usually make these errors mostly due to overgeneralizations and simplifications (Heydari and Bagheri, 2012). The learner will overgeneralize and apply the grammatical rule to a place where it does not apply. For simplification, the learner will omit the rule in the context when the rule is supposed to apply. Most of this is due to the learner not having a frame of reference for that rule in their native language. For example, Chinese does not use article or determiners so they tend to overgeneralize or simplify and some times insert or omit an article or determiner (Robertson, 2000). Table 1 shows examples of a syntactic error and a semantic error. In the syntactic error example, the use of the form of the verb is incorrect. *Working* should be changed to *to work*. In the semantic example, *Lately* should be changed to *Recently*.

For the unrestricted track, we created a pseudo corpus by using the syntactic errors from the dictionary described in the previous section. We used 6 types of syntactic errors based on the ERRANT annotation (Bryant et al., 2017), including ADJ:FORM (*is **good** for our health than* – *is **better** for our health than*), MORPH (*the everyday **invents*** – *the everyday **inventions***), NOUN:INFL (***TVs** companies* – ***TV** companies*), VERB:FORM (*make my dream **comes** true* – *make my dream come true*), VERB:INFL (*he **thinked*** – *he **thought***) and VERB:SVA (*there **are** a* – *there **is** a*). The total number of syntactic error pairs we extracted from the dictionary is around 100K entries. The clean corpus we use has around 2 million sen-

[1] From *English Gigaword*, https://catalog.ldc.upenn.edu/LDC2003T05

Syntactic Error:	*I want **working** in our cafe.*
Semantic Error:	***Lately** I have seen a very interesting TV show.*

Table 1: Examples of a syntactic error and a semantic error

tences from the AFP news corpus. Each error in the syntactic error dictionary can be used at most once. We keep the numbers of different types of errors the same and thirty percent of the sentences in the pseudo learner corpus contains exactly one error. For each sentence, the search of the phrase starts from the longest length. Once we find an n-gram that appeared in the correct side of the dictionary, we replace it with the incorrect counterpart. The cheat sheet and the pseudo syntactic error corpus improved our results by emphasizing the learners' errors and their contexts.

4 Experiments and Results

4.1 Experiment settings

We used all the four datasets — FCE, NUCLE, W&I+LOCNESS and Lang-8 — provided in the BEA 2019 Shared Task[2] as our base data. For the restricted track, we appended the data with cheat sheet (totally over 2M sentences), and for the unrestricted track, the additional syntactic pseudo corpus was supplemented (over 4M sentences in total). The official W&I+LOCNESS development set and test set were used as development and evaluation[3]. We did not use any spell check to pre- or post-process our data, which could affect our results negatively (Chollampatt and Ng, 2017).

For the SMT approach to GEC, we used the same Moses (Koehn et al., 2007) setup as in §2.2, except for the sentence length, which we changed to the default value (1 – 80). The standard Minimum Error Rate Training (MERT) algorithm (Och, 2003) was used for tuning. For the NMT approach, we used a 7-layer convolutional seq2seq model[4] as described in Chollampatt and Ng (2018) with similar hyper-parameters, such as the top 30K BPE tokens as the input and output vocabularies, 1,024 (hidden size) × 3 (convolution window size) in the encoders and decoders, Nesterov Accelerated Gradient as the optimizer with a momentum of 0.99, dropout rate of 0.2 and an adaptive learning rate (initially 0.25, minimum 10^{-4}). Differently from Chollampatt and Ng (2018), we set the word embedding dimensions in both encoders and decoders to 300 rather than 500, and we trained the word embeddings separately using the error and correct side training data instead of external corpora. During inference, we used a beam size of 10.

		Prec.	**Recall**	**$F_{0.5}$**
Restricted	SMT	51.48	17.85	37.39
	NMT	**63.31**	15.43	**39.06**
Unrestricted	SMT	56.03	15.85	37.18
	NMT	**65.14**	17.63	**42.33**

Table 2: Results submitted to the BEA 2019 Shared Task

4.2 Results

Table 2 shows the results we submitted to the BEA 2019 Shared Task. The results are all from the versions with an error+context cheat sheet because our phrase table cheat sheet yielded much worse results. Overall, our models with error+context cheat sheet achieved comparably high precision in both restricted and unrestricted tracks. Within our own models, GEC-tuned NMT, as expected, consistently outperformed the generic SMT model. In the unrestricted setting, for example, the gap in $F_{0.5}$ was over 5%. When comparing the two NMT models across the two tracks, our results clearly show that the additional pseudo corpus contributed to better performance in precision, recall and $F_{0.5}$.

5 Conclusion and Future Work

In this study, we explored two error-focused approaches to grammatical error correction. One was to extract parallel error-correct phrases (error + surrounding context) from the GEly. Extracting error phrases is not a new method per se, as previous studies have used these phrases tC corpora and append them to our training data directo generate artificial errors (e.g., Felice, 2016). However, we purposefully included these phrases in our training in order for our models to pay attention to these errors and to focus on one error at a time. As ex-

[2] https://www.cl.cam.ac.uk/research/nl/bea2019st/

[3] https://competitions.codalab.org/competitions/21922

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

pected, the precision of our GEC models improved significantly, and there was at least a 2% increase in $\mathrm{F}_{0.5}$.

The second approach was to incorporate phrases from SMT-generated phrase translation table. In the current study, we extracted parallel phrases with five or more words and the direct translation probability (from error to correct) over 95%. Contrary to our prediction, appending these phrases to our training data significantly decreased the performance. A closer examination of the phrases shows that there are many partial redundancies, which may have caused our models to miss focus. Thus, we plan to investigate various techniques to prune the phrase table (e.g. Johnson et al., 2007; Zens et al., 2012) so that errors are truly highlighted as in the error+context approach.

In the unrestricted track, we injected syntactic errors from our error+context dictionary to a clean corpus and appended the artificial error corpus to the training data for the restricted track. When training with SMT, there was no performance gain overall and at the syntactic error type level. For example, the precision of the VERB:FORM error type was only 48.98% and the $\mathrm{F}_{0.5}$ was 35.40%. However, when the same data was trained with NMT, the benefit of additional data was evident. The precision and $\mathrm{F}_{0.5}$ of VERB:FORM almost doubled in this setting, compared to that in SMT. These results, again, demonstrate the limitations of the generic SMT approach to grammatical error approach (e.g. Yuan and Felice, 2013).

The recall of our models stayed low across all the settings, which indicates our models were too conservative. The conservativeness can be mainly attributed to the large proportion of unchanged sentences in the training data. Indeed, our pseudo corpus generation process was constrained as only 30% of the two million sentences were applied error injection. We will further explore the relationship between recall and proportion of unchanged sentences in GEC.

Finally, our current study only focused on syntactic errors, which should be easier for MT models to detect and correct compared to semantic errors, because semantic errors require knowledge about meaning in addition to structure. Given the complexity of language, the individual meaning of a word in a sentence changes according to the context. A simple example, *She kicked the bucket.* and *He filled the bucket with soda.* both contain the word *bucket*, but the meanings are drastically different. Traditional word embeddings such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017) only have one representation per word. As the meanings of each word changes based on the surrounding context, in which previous methods fail. Therefore, we require a model that is capable of understanding the variations in meanings of the given word based on its surrounding text in the sentence. ELMo is another method for text embedding (Peters et al., 2018) which uses a deep, bi-directional LSTM model that takes contextual information into account and achieves state-of-the-art results in many NLP tasks. ELMo analyses words within the context that they are used, hence the way ELMo is used is quite different to word2vec or fastText. As opposed to having a dictionary of words and their corresponding vectors, ELMo instead creates vectors on-the-fly by passing text through the deep learning model. The model is character based and hence forms representations of out-of-vocabulary words. We will investigate whether incorporating ELMo in our NMT model can improve the performance of correcting semantic errors in the near future.

References

Nguyen Bach. 2012. Dependency structures for statistical machine translation. *ProQuest LLC*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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 and Hwee Tou Ng. 2017. Connecting the dots: Towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, Copenhagen, Denmark. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, New Orleans, Louisiana.

Mariano Felice. 2016. Artificial error generation for translation-based grammatical error correction. Technical Report UCAM-CL-TR-895, University of Cambridge, Computer Laboratory.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency Boost Learning and Inference for Neural Grammatical Error Correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Pooneh Heydari and Mohammad S Bagheri. 2012. Error analysis: Sources of l2 learners' errors. *Theory & Practice in Language Studies*, 2(8).

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving Translation Quality by Discarding Most of the Phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic. Association for Computational Linguistics.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. 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, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *http://arxiv.org/abs/1301.3781*.

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.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. 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 2227–2237, New Orleans, Louisiana. 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.

Daniel Robertson. 2000. Variability in the use of the english article system by chinese learners of english. *Second language research*, 16(2):135–172.

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.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983, Jeju Island, Korea. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data (to be appeared in NAACL2019).