Grammatical Error Correction via Supervised Attention in the Vicinity of Errors

Hiromichi Ishii[1*] **Akihiro Tamura**[2] **Takashi Ninomiya**[1]
[1]Ehime University [2]Doshisha University
ishii@ai.cs.ehime-u.ac.jp
aktamura@mail.doshisha.ac.jp ninomiya@cs.ehime-u.ac.jp

Abstract

This study proposes a new supervised attention mechanism for grammatical error correction, which is trained to focus the encoder-decoder attention to each word in the corrected sentence on words adjacent to corresponding words in the error sentence. Experiments on the CoNLL 2014 test set show that the performance of a Transformer-based grammatical error correction model, Copy-Augmented Transformer, can be improved by 0.73 $F_{0.5}$ points by incorporating our proposed attention mechanism.

1 Introduction

Extensive research has taken place on grammatical error correction techniques which automatically correct grammatically erroneous sentences (error sentences) into accurate sentences (corrected sentences), due in part to their usefulness as tools for foreign language learning. A variety of methods have been proposed in the past. In recent years, methods using neural networks have achieved the highest accuracy and have become mainstream in the field. Among grammatical error correction models using neural networks, those based on Transformer (Vaswani et al., 2017) architecture have performed well. Transformer-based models' characteristics include a self-attention mechanism that grasps relationships between words within the same sentence (error sentence or corrected sentence), as well as an encoder-decoder attention mechanism that identifies the error sentence words on which to focus when producing the words of the corrected sentence (relationship between the words of the error sentence and those of the corrected sentence). In general, the relationships between words captured by these attention mechanisms are learned automatically.

Chollampatt and Ng (2018) compare and analyze encoder-decoder attention automatically captured via grammatical error correction models based on CNN and LSTM. They consider the attention given to each word in the corrected sentence, concluding that focusing on the words adjacent to corresponding words in the error sentence may be more beneficial than focusing on corresponding words in the error sentence themselves.

Using a Copy-Augmented Transformer (Zhao et al., 2019) — a model of grammatical error correction based on Transformer architecture — this study proposes a method of machine learning whereby one of the heads of a multi-head encoder-decoder attention mechanism is programmed to focus the attention given to each word in the corrected sentence only on words adjacent to corresponding words in the error sentence. Through the proposed method, the model is expected to be able to learn a grammatical error correction model by focusing on error-adjacent words.

An evaluation experiment on grammatical error correction was conducted using the CoNLL-2014 test-set (Ng et al., 2014). A Copy-Augmented Transformer was programmed to focus its attention only on words adjacent to corrected words, resulting in a confirmed increase in $F_{0.5}$ by 0.73 points.

*Present affiliation is Daihatsu Motor Co., Ltd.

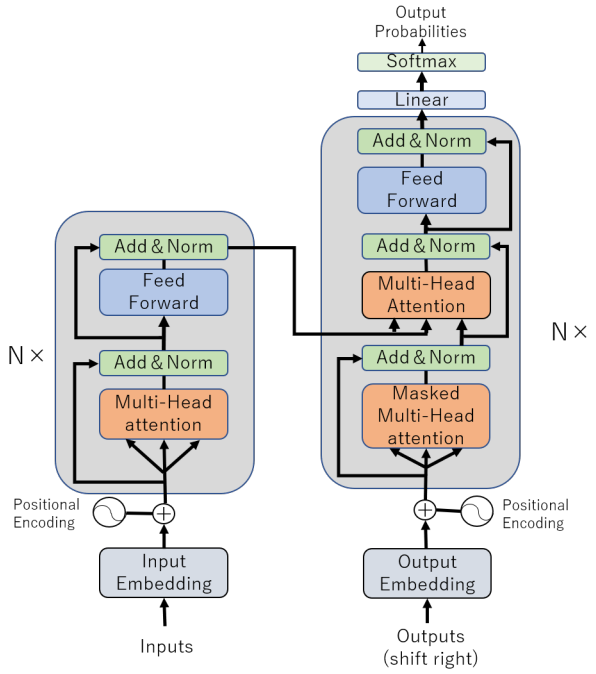

Figure 1: Schematic diagram of the Transformer

2 Copy-Augumented Transformer

This section illustrates the Copy-Augmented Transformer (Zhao et al., 2019) on which the proposed model is based. This model achieves high error correction performance by introducing a Copy Mechanism and a pre-training method into the Transformer model.

2.1 Transformer

A Transformer is an encoder-decoder model comprising an encoder that transforms an input sequence $X = (x_1, x_2, \cdots, x_n)$ into an intermediate representation $Z = (z_1, z_2, \cdots, z_n)$ and a decoder that transforms the intermediate representation Z into an output sequence $Y = (y_1, x_2, \cdots, y_m)$. A schematic diagram of the transformer is shown in Figure 1.

The encoder-decoder is composed of a stack of N individual encoder layers and decoder layers. Each encoder layer is composed of two sublayers: a multi-head self-attention mechanism and a position-wise fully connected layer. The decoder layer is composed of three sublayers: a multi-head self-attention mechanism, a position-wise fully connected layer, and a multi-head encoder-decoder attention mechanism. Residual connection and normalization occur between sublayers. Each head in the multi-head self-attention and multi-head encoder-decoder attention mechanisms calculates its attention according to Equation 1, using a scaled dot-product attention mechanism.

$$\begin{aligned} Attention(Q, K, V) &= AV, \\ A &= \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}}), \end{aligned} \tag{1}$$

where A is called *attention matrix* and the elements of A are called *attention weights*, Q, K and V are hidden states of an encoder/decoder, and d_k is the number of dimensions of Q, K and V. Q, K and V stand for query, key and value. In the scaled dot-product attention mechanism, dot-products of Q and K calculates the degree of association between elements. A softmax function is applied to the resulting values, which are then multiplied by V to calculate the weight sum value of the elements, i.e. their relative strength and importance. In the self-attention mechanism, a single input source (the hidden state of the encoder or the decoder) is used for Q, K, and V, making it possible to calculate the relative strength of words within the same sentence. In the encoder-decoder attention mechanism, the most recent hidden state of the decoder is used as Q, and the hidden states of the encoder are used as K and V, making it possible to calculate the strength of the relationship between single words in the input sentence and output words. Then the representations yielded by each head are concatenated and converted into embedded dimensions via linear transformation.

$$MultiHead(Q, K, V) = Concat(h_1, ..., h_h)W_O \tag{2}$$

$$h_i = Attention(Q_i, K_i, V_i) \tag{3}$$

Here, W_O is the weight matrix.

The output of the Nth decoder layer is converted into a vocabulary-size dimension via linear transformation, and an output word probability distribution is calculated by applying the softmax function. Then the output sentence is generated on the basis of the probability distribution. The loss function for the

training data is expressed as

$$-\sum_{i=1}^{D} \log P(Y^i \mid X^i), \tag{4}$$

where D is the size of the training data, and the i-th training data is (X^i, Y^i).

2.2 Copying Mechanism

Zhao et al. (2019) introduce the Copying Mechanism (See et al., 2017) and the pre-training method to the Transformer model for grammatical error correction. The Copying Mechanism is a mechanism that sequentially determines, when the decoder is generating a sentence, whether words are to be copied from the input sentence or generated anew. Specifically, the mechanism determines each output word at sequence point t according to the probability calculated using Equation 5 below.

$$\begin{aligned} P(y_t|X) =&(1-\alpha_t^{copy})P^{gen}(y_t|X)+ \\ &\alpha_t^{copy}P^{copy}(y_t|X) \end{aligned} \tag{5}$$

P^{gen} is the probability of an output word generated by the decoder of the Transformer; P^{copy} is the probability of an output word copied from the input sentence; and α_t^{copy} is the weight of the adjustment of either generating or copying the word. P^{copy} and α_t^{copy} are calculated on the basis of encoder-decoder attention as shown below.

$$q_t = h_t^{trg}W_Q^T,\ K = H^{src}W_K^T,\ V = H^{src}W_V^T \tag{6}$$

$$A_t = q_t^T K \tag{7}$$

$$P^{copy}(y_t|X) = softmax(A_t) \tag{8}$$

$$\alpha_t^{copy} = sigmoid(W^T \sum (A_t^T \cdot V)) \tag{9}$$

Here, h_t^{trg} is the hidden state of the decoder of time step t, H^{src} is the hidden state sequence of the encoder $(h_1^{src}, ..., h_n^{src})$, and W_Q, W_K, W_V, and W are the weight matrices of each.

2.3 Pre-training

Pre-training was conducted using a noise-reduction self-encoder to augment training data quantity, following Zhao et al. (2019). Specifically, One Billion Word Benchmark data (Chelba et al., 2013) sentences were used to create pseudo-error sentences by eliminating sentence words with 10% probability, inserting words with 10% probability, replacing sentence words with dictionary words with 10% probability, and changing positional relationships between words with 70% probability. The copy-augmented Transformer parameters were then trained using these pseudo-error sentences.

3 Proposed Method

This section proposes a method whereby one of the heads of the multi-head encoder-decoder attention mechanism of the Transformer is trained to constrain each word in the corrected sentence to attend to words adjacent to the corresponding word in the error sentence. Figure 2 is an example of the creation of supervision data for learning the constraints on encoder-decoder attention.

The proposed method first analyzes word alignment between words in the corrected sentence and error sentence using an alignment tool (Process 1). Next, using the alignment results of Process 1, it identifies for each word in the corrected sentence the two words that precede and the two words that follow its corresponding word in the error sentence (Process 2). Finally, it assigns an attention weight of "1/number of identified words" to each word identified in Process 2, and an attention weight of 0 to the other words in the error sentence (Process 3). For example, in Figure 2, the word "to" in the corrected sentence corresponds to the single word "in" in the error sentence, where the two preceding and following words are the four words "I," "went," "Tokyo," and "by." Therefore, each of these four words is assigned an attention weight of $0.25(= 1/4)$, whereas the other words in the error sentence ("Yesterday," ",," "in," "bus" and ".") have an attention weight of 0.

The grammatical error correction model is trained using the attention weight thus obtained for each word in the corrected sentence as supervision data for learning the attention matrix in Equation 1 in one of the heads of the multi-head encoder-decoder attention mechanism. Specifically, when training the grammatical error correction model, the following loss function, which introduces the constraints for

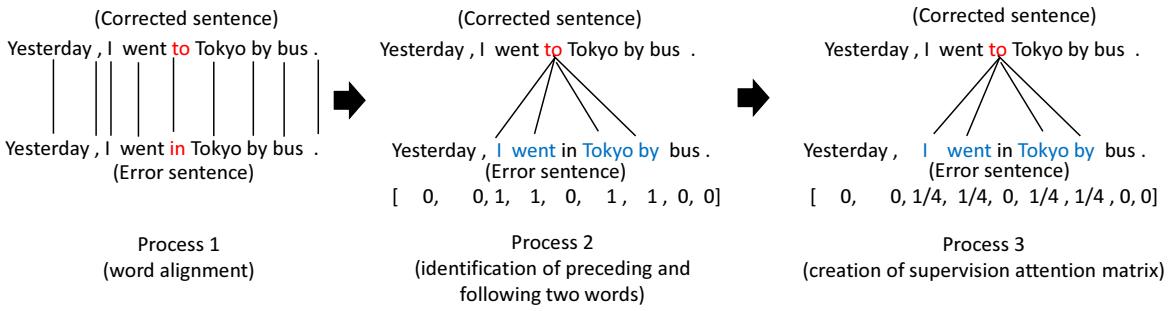

Figure 2: Example of creation of training data for learning the constraints on encoder-decoder attention.

Method	Precision(%)	Recall(%)	$F_{0.5}$(%)
Baseline Model	68.30	37.93	58.87
Proposed Method	69.10	38.02	**59.60**
Chollampatt and Ng (2018)	60.9	23.7	46.4
Junczys-Dowmunt et al. (2018)	-	-	53.0
Grundkiewicz and Junczys-Dowmunt (2018)	66.8	34.5	56.3
Junczys-Dowmunt et al. (2018)	65.5	37.1	56.8
Kiyono et al. (2019)	67.9	44.1	61.3
Omelianchuk et al. (2020)	77.5	40.1	65.3
Lichtarge et al. (2020)	69.4	43.9	62.1
Kaneko et al. (2020)	69.2	45.6	62.6
Wan et al. (2020)	69.5	47.3	63.5
Stahlberg and Kumar (2021)	72.8	49.5	66.6
Zhao et al. (2019) (Baseline model article value)	68.97	36.98	58.80
Zhao et al. (2019) (Baseline model + multi-task article value)	67.74	40.62	59.76

Table 1: Experiment Results

the encoder-decoder attention mechanism, is used.

$$-\sum_{i=1}^{D} \log P(Y^i \mid X^i) + \lambda \Delta(A^i, \hat{A}^i) \qquad (10)$$

Here, λ is the hyper-parameter and Δ is the error between the word correspondence relationship captured via the encoder-decoder attention matrix A^i and the word correspondence relationship provided as supervision for attention matrix $\hat{A}^i$. Δ is calculated as follows.

$$\Delta(A^i, \hat{A}^i) = -\sum_{m}\sum_{n} \hat{A}^i{}_{m,n} \log A^i_{m,n} \qquad (11)$$

Here, $\hat{A}^i_{m,n}$ and $A^i_{m,n}$ are attention weights that express the relationship between m-th word in the corrected sentence and n-th word in the error sentence. Specifically, $\hat{A}^i_{m,n}$ equals the attention weight found through the encoder-decoder attention constraints of Process 3, and $A^i_{m,n}$ equals the attention weight as in the sum weight value calculated by the encoder-decoder attention mechanism via Equation 1.

4 Experiment

The experiment used approximately 1.2 million sentences from the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), Lang-8 learner Corpus (Mizumoto et al., 2011), and FCE (Yannakoudakis et al., 2011) as training data. CoNLL-2013 test data (Dahlmeier et al., 2013) was used as development data, and the CoNLL-2014 test-set was used as evaluation data. The pre-processing of each dataset followed Zhao et al. (2019).

This experiment conducted a comparative perfor-

Error Sentence	Sometimes some family structure and cultural beliefs can influence the pattern of communication .
Corrected Sentence	Sometimes some family structures and cultural beliefs can influence the pattern of communication .
Baseline Model	Sometimes family structure and cultural beliefs can influence the pattern of communication .
Proposed Method	Sometimes some family structures and cultural beliefs can influence the pattern of communication .

Table 2: Example of Precision Improvement

Error Sentence	Some people fake their identities in media sites so that they can know more people or some may even cheat others .
Corrected Sentence	Some people fake their identities on media social sites so that they can know more people or some may even deceive others .
Baseline Model	Some people fake their identities in media sites so that they can know more people or some may even cheat others .
Proposed Method	Some people fake their identities on media sites so that they can know more people or some may even cheat others .

Table 3: Example of Recall Improvement

mance analysis of the proposed method using the Copy-Augmented Transformer (Zhao et al., 2019) as a baseline. The baseline model used was implemented by Zhao et al.[1]. The proposed method is a model trained applying the constraints of encoder-decoder attention to the baseline explained in Section 3. Error correction performance was evaluated according to $F_{0.5}$ values calculated via MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012). $F_{0.5}$ is standardly used for evaluating grammatical error correction performance because precision is considered more important than recall for grammatical error correction.

Both the baseline model and the proposed model had 6 stacks of encoder and decoder layers, featured 8 heads, and used $d_{model} = 512$ and $d_{ff} = 4096$ as hidden state dimensions. Further, both used the Nesterovs Accelerated Gradient for optimization and set the learning rate at 0.02, the weight decay at 0.5, and the edit-weighted MLE at $\Lambda = 1.2$. During decoding, a beam size of 12 was used. The proposed method set the λ hyper-parameter at 0.05, and introduced conditions to the encoder-decoder attention mechanism of the 5th layer, following (Garg et al., 2019). Further, when introducing the conditions, it used GIZA++ as an alignment tool to create the training data.

The results of the experiment are shown in Table 1. The table shows CoNLL-2014 test-set accuracy scores for the baseline model and the proposed method, as well as other existing methods that do not use ensembles. In Table 1, $F_{0.5}$ for the proposed method is 0.73 points higher than the baseline. This result empirically confirms that error correction performance can be improved by training an encoder-decoder attention mechanism to focus the attention of each word in the corrected sentence on words adjacent to corresponding words in the error sentence. The table also introduces accuracy scores for other existing methods that do not use ensembles as further points of comparison. The proposed method and the baseline of this study do not use multi-task learning. Therefore, they both score lower than the multi-task learning method proposed by Zhao et al. (2019).

5 Discussion

The experiment compared correction accuracy for the baseline model and the proposed method. The

[1] https://github.com/zhawe01/fairseq-gec

results empirically confirm a 0.8-point increase in precision and a 0.09-point increase in recall. These results succeed in demonstrating that the introduction of the proposed constraints determines an increase in grammatical error correction accuracy and error position detection.

The increase in error correction precision may be due to a decrease in unnecessary corrections of already correct points, because the encoder-decoder attention mechanism is programmed, when generating the corrected sentence, to identify the adjacent words in addition to the error itself in the error sentence, thus becoming able to determine whether or not to make a correction while taking into account the adjacent words. Table 2 shows an example of this. The error sentence in Table 2 is "some family structure, " and the error position is "structure." With the baseline model, "some" is identified as an error and corrected, whereas with the proposed method, the correct sentence "some family structures" is generated by taking into account the word "some," as confirmed by the experiment. This result is likely due to the proposed model becoming able to take into account adjacent words when generating correction words.

The increase in error correction recall is probably caused by an improved ability to detect error positions that could not be identified by focusing only on erroneous words (e.g. articles or prepositions). Table 3 shows an example of this. In Table 3, "in" must be corrected to "on," to produce the correct sentence. The preposition error, which remains uncorrected in the baseline model, is corrected with the proposed method, as confirmed by the experiment. This is likely due to the fact that, whereas the baseline model devotes more attention to the erroneous words due to automated learning of the encoder-decoder attention mechanism, the proposed method is able to identify preposition errors on the basis of data from adjacent words, such as "media sites" in this case.

6 Related Work

Supervised learning of attention has been recently studied in various NLP tasks such as semantic role labeling (Strubell et al., 2018), word alignment (Garg et al., 2019) and machine translation (Deguchi et al., 2019; Bugliarello and Okazaki, 2020). They basically use automatic NLP tools, such as dependency parsers for learning self-attention and word alignment tools for learning encoder-decoder attention, for generating supervision data because annotating dependencies or word alignment is a laborious and expensive task. Attention is trained as multi-task learning by minimizing the difference between an attention matrix and its supervision data in the loss function.

As a closely related work to ours, Garg et al. (2019) proposed supervised encoder-decoder attention for word alignment and machine translation by using word alignment tools for generating supervision data. The difference between our method and theirs is that (i) their task is different from our task and (ii) their attention mechanism is trained to attend to the aligned source word, but ours is trained to attend to the positions in the vicinity of the aligned source words. We consider that grammatical error correction, unlike machine translation, can be improved by training an encoder-decoder attention mechanism to focus the attention of each word in the corrected sentences on words adjacent to the aligned words in the error sentence.

7 Conclusion

This study proposes a method whereby an encoder-decoder attention mechanism is trained to focus the attention of each word in the corrected sentence on words adjacent to corresponding words in the error sentence. It empirically confirms that $F_{0.5}$ values can be improved by 0.73 points by training one of the heads of the multi-head encoder-decoder attention mechanism of a Copy-Augmented Transformer with the proposed constraints. Going forward, the efficacy of the proposed method must be tested against other datasets and grammatical error correction models.

References

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online, July. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

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*.

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, June. 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, June. Association for Computational Linguistics.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-based self-attention for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 239–246, Varna, Bulgaria, September. INCOMA Ltd.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. 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 4453–4462, Hong Kong, China, November. Association for Computational Linguistics.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 284–290, New Orleans, Louisiana, June. 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, June. 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, July. 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, November. 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, 10.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

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, June. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online, July. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with

tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium, October-November. 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.

Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. Improving grammatical error correction with data augmentation by editing latent representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online), December. International Committee on 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, June. 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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota, June. Association for Computational Linguistics.