

Grammatical Error Correction as GAN-like Sequence Labeling

Kevin Parnow^{1,2,3,†}, Zuchao Li^{1,2,3,†}, and Hai Zhao^{1,2,3*}

¹Department of Computer Science and Engineering, Shanghai Jiao Tong University

²Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China

³MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

{parnow, charlee}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

In Grammatical Error Correction (GEC), sequence labeling models enjoy fast inference compared to sequence-to-sequence models; however, inference in sequence labeling GEC models is an iterative process, as sentences are passed to the model for multiple rounds of correction, which exposes the model to sentences with progressively fewer errors at each round. Traditional GEC models learn from sentences with fixed error rates. Coupling this with the iterative correction process causes a mismatch between training and inference that affects final performance. In order to address this mismatch, we propose a GAN-like sequence labeling model, which consists of a grammatical error detector as a discriminator and a grammatical error labeler with Gumbel-Softmax sampling as a generator. By sampling from real error distributions, our errors are more genuine compared to traditional synthesized GEC errors, thus alleviating the aforementioned mismatch and allowing for better training. Our results on several evaluation benchmarks demonstrate that our proposed approach is effective and improves the previous state-of-the-art baseline.

1 Introduction

Sequence-to-sequence neural solutions (Parnow et al., 2020) have been quite successful in comparison to their statistical counterparts (Sutskever et al., 2014), but these approaches suffer from a couple key problems, which has given rise to sequence labeling approaches for GEC (Omelianchuk et al.,

2020). Such approaches task models with generating a list of labels to classify the grammatical errors in a sentence before correcting these errors.

Sequence labeling approaches have recently gained popularity in GEC and are currently state-of-the-art. One typical aspect of sequence labeling approaches is labeling and correcting sentences through an iterative process. As successive edits will depend on how other errors are corrected in a sentence, using an iterative process and correcting only the most salient errors in each round allows models to achieve better performance; however, because of this process, models are tasked with handling sentences with varying rates of errors, as during each round of inference for a given sentence, a model encounters a sentence with progressively fewer errors. This of course causes an exposure bias problem, as the training data does not match the test data, and suggests that providing the model with training data with varying error rates will lead to better performance.

To combat this exposure bias, we propose a new approach for training a sequence labeling GEC model that draws from GANs (Goodfellow et al., 2014), which consist of a generator that generates increasingly realistic fake inputs and a discriminator that is tasked with differentiating these fake inputs from real inputs. Other GEC works like (Raheja and Alikaniotis, 2020) directly used GANs to produce grammatically correct sentences given grammatically incorrect ones. This contrasts our work, which uses aspects of a GAN to enhance the training process rather than using a GAN itself as the correcting model. Our model consists of three components: an encoder, a Grammatical Error Detector, and a Grammatical Error Labeler. By sampling from the error distribution in the error labeler, our model can synthesize sentences with new errors creating new sentence pairs for further training data. As a result, our Detector continually

*Corresponding author. † These authors made equal contribution. This work was supported by Huawei Noah's Ark Lab and funded by the National Key Research and Development Program of China (No. 2017YFB0304100), the Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), the Huawei-SJTU long term AI project, Cutting-edge machine reading comprehension and language model.

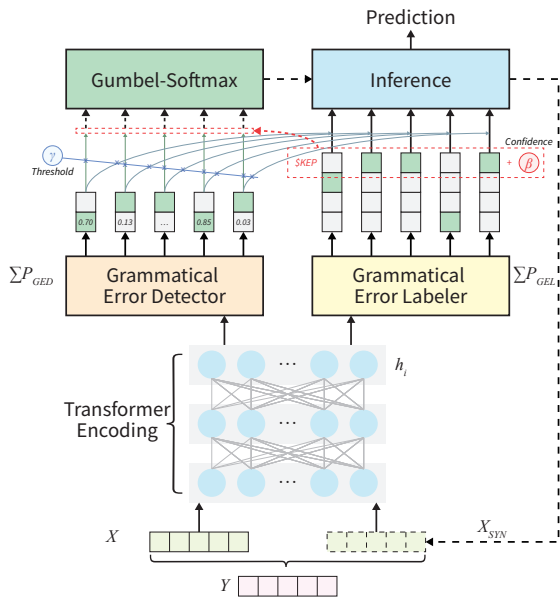


Figure 1: An overview of our model.

improves its ability to detect errors and essentially acts as a discriminator of errors, and our Labeler continually improves the authenticity of its error distribution and becomes a better generator of errors. This process allows us to counter the exposure bias problem sequence labeling GEC models face because in addition to allowing us to generate new errorful sentences whose errors are increasingly representative of those in real data, we can also use control parameters to set the error rates of these sentences and accommodate our iterative inference process.

2 Our Approach

We formulate the GEC task as a problem of sequence labeling and create a neural sequence labeling model based on a deep pre-trained Transformer encoder to deal with this problem. Inspired by the work of (Omelianchuk et al., 2020), our full model’s overall architecture diagram is shown in Figure 1. There are three main components in our basic neural GEC model: a deep pre-trained Transformer Encoder, a Grammatical Error Detector, and a Grammatical Error Labeler. To accommodate our new GAN-like training process, we add a Gumbel-softmax sampling component to the basic GEC model.

2.1 Background and Notation

First, in training, given incorrect input sentence $X = x_1, x_2, \dots, x_n$ and its corrected version

$X_c = y_1, y_2, \dots, y_m$, the model predicts a corrective label sequence $T = t_1, t_2, \dots, t_n$ by minimizing the token-level Levenshtein distance on the span-based alignments of X and X_c . The corrective label set is given as $\mathcal{T} = \{\$KEP, \$DEL, \$APP, \$REP\} \cup \{\$CAS, \$MRG, \$SPL, \$NNUM, \$VFORM\}$, in which the first set consists of the basic text editing transformation operations and the second consists of g -transformations as defined by (Omelianchuk et al., 2020) for GEC¹. Aligning sentences using these transformations in pre-processing, reduces what would be a sequence generation task that handles unequal source-target lengths to a set of label classification problems. In this formulation, the neural sequence labeling model trains to optimize the input sequence’s negative log-likelihood loss for an input sequence:

$$\mathcal{J}(\theta) = - \sum_{i=1}^n \log p(t_i|x, \theta),$$

where p is the conditional probability that the model outputs at each position i .

2.2 Deep Pre-trained Transformer Encoder

As in most neural sequence labeling models (Ma and Hovy, 2016), a neural encoder such as a BiLSTM (Hochreiter and Schmidhuber, 1997) or a Transformer (Vaswani et al., 2017; Li et al., 2021) is used to extract context-aware features from the input sequence. Deep pre-trained language models such as BERT (Devlin et al., 2019; Zhang et al., 2020b), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019) have recently demonstrated the efficacy of Transformer models trained on large-scale unlabeled data in various NLP tasks. We leveraged these very beneficial models by using a pre-trained language model as our encoder. We define the contextualized features captured by the neural encoder as:

$$h_i = [\mathbf{Enc}(X)]_i,$$

where \mathbf{Enc} represents the encoder, and $[\cdot]_i$ represents the output of the i -th position after encoding.

2.3 Grammatical Error Detector and Labeler

Next, we adopt a Grammatical Error Detector (GED) to detect the presence of errors and a Gram-

¹The label set here only presents the transformations’ basic names. Some transformations require additional parameters because they are context-specific and thus have many different versions.

matical Error Labeler (GEL) to predict detailed error labels. With these labels, corrections are applied to sentences, and this process is typically iterative, as some corrections may depend on others, and applying corrections only once may not be enough to fully correct the sentence. During iterative correction, the model needs to assess at each round whether more correction is required. To this end, we use the GED to determine the degree of error for an entire sentence and control the iterative correction process.

Specifically, we use a binarization Y_b of the corrective labels Y as the training target of the GED and use Y as the training target of the GEL. To obtain label probabilities grammatical error detection and labeling, two linear layers with softmax layers are appended to the encoder:

$$P_{\text{GED}}^i = \text{softmax}(\text{MLP}_{\text{GED}}(h_i)),$$

$$P_{\text{GEL}}^i = \text{softmax}(\text{MLP}_{\text{GEL}}(h_i)).$$

The binary classification probabilities in the GED output do not necessarily control the inference process’s iterations. Rather, after using the GEL error label probabilities as thresholds for sentence positions, we also use the sum of these probabilities as a threshold for attempting another round of correction on the whole sentence. The model continues correcting the sentence until either it reaches a preset maximum number of iterations or no longer satisfies the following condition: $\sum_i [P_{\text{GED}}^i]_{err=1} > \gamma$, where γ is the minimum error probability threshold for a sentence.

Additionally, since GEC usually corrects a small portion of a sentence (and there are therefore no errors in most of the input), the corrective label prediction task is an imbalanced classification problem. We alleviate this imbalance classification issue by taking advantage of this prior knowledge and adding a fixed and preset confidence β to the label \$KEP to keep a position unchanged when applying corrections:

$$[P_{\text{GEL}}^i]_{\text{\$KEP}} = [P_{\text{GEL}}^i]_{\text{\$KEP}} + \beta.$$

2.4 GAN-like Sequence Labeling Training

While we adopt sequence labeling instead of sequence-to-sequence modeling in this paper and therefore avoid the exposure bias problem caused by left-to-right sequence generation, our model still faces exposure bias because of the iterative correction process, which, through its iterative correction

process, tasks the model with handling much more varied error rates in inference compared to in training, where it handles static data and does not use multiple-round corrections. To address this issue, we borrow the idea of a GAN (Goodfellow et al., 2014) and propose a GAN-like iterative training approach for a sequence labeling GEC model. GANs, whose training objective can be formulated as a minimax game between a generator that creates increasingly realistic fake outputs and a discriminator that must differentiate these outputs from their real counterparts, have been suggested for sequence-to-sequence text generation (Li et al., 2020; Zhang et al., 2020a; Li et al., 2018) as they do not suffer from exposure bias.

Algorithm 1 GAN-like Sequence Labeling Training

Require: Genuine GEC parallel dataset $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y})\}$
Synthesized GEC parallel dataset $\mathcal{D}_{\text{SYN}} = \{\}$
Number of training stages \mathcal{N}
Number of training epochs M
Sentence error probability threshold γ
Additional confidence β for label \$KEP

- 1: **for** i in 1, ..., N **do**
- 2: Initialize model parameters from previous training stage $\theta_i \leftarrow \theta_{i-1}$ when $i > 1$
- 3: **for** j in 1, ..., M **do**
- 4: **for** k in 1, ..., $|\mathcal{D} \cup \mathcal{D}_{\text{SYN}}|$ **do**
- 5: Encode each sentence \mathbf{X}_k as \mathbf{H}_k
- 6: $P_{\text{GED}}^k = \text{Softmax}(\text{MLP}_{\text{GED}}(\mathbf{H}_k))$
- 7: $P_{\text{GEL}}^k = \text{Softmax}(\text{MLP}_{\text{GEL}}(\mathbf{H}_k))$
- 8: $loss_{\text{GED}} = \text{CrossEntropy}(P_{\text{GED}}^k, \mathbf{Y}_{err}^k)$
- 9: $loss_{\text{GEL}} = \text{CrossEntropy}(P_{\text{GEL}}^k, \mathbf{Y}_{label}^k)$
- 10: $loss = loss_{\text{GED}} + loss_{\text{GEL}}$
- 11: Update the model parameter θ_i with $loss$
- 12: **end for**
- 13: **end for**
- 14: $\mathcal{D}_{\text{SYN}} = \{\}$
- 15: **for** k in 1, ..., $|\mathcal{D}|$ **do**
- 16: Encode each sentence \mathbf{X}_k as \mathbf{H}_k
- 17: $P_{\text{GED}}^k = \text{Softmax}(\text{MLP}_{\text{GED}}(\mathbf{H}_k))$
- 18: $P_{\text{GED}}^k = \sum [P_{\text{GED}}^k]_{err=1} > \gamma$
- 19: $P_{\text{GEL}}^k = \text{Softmax}(\text{MLP}_{\text{GEL}}(\mathbf{H}_k))$
- 20: $[P_{\text{GEL}}^k]_{\text{\$KEP}} = [P_{\text{GEL}}^k]_{\text{\$KEP}} + \beta$
- 21: $P_{\text{GEL}}^k = \text{GumbelSoftmax}(P_{\text{GEL}}^k)$
- 22: Use P_{GED}^k and P_{GEL}^k to produce sampled sequence $\mathbf{X}_{\text{SYN}}^k$
- 23: $\mathcal{D}_{\text{SYN}} = \mathcal{D}_{\text{SYN}} \cup \{(\mathbf{X}_{\text{SYN}}^k, \mathbf{Y}_k)\}$
- 24: **end for**
- 25: **end for**

In our model, the GEL module can be considered a discriminator, as it must differentiate whether tokens are erroneous, and by adding a sampling module to the GED module, we can create a generator that outputs grammatical errors (rather than corrections) that are increasingly realistic. We can then pair these sampling outputs with their golden sequence in the training dataset to create new training

GEC system	Ens.	CoNLL-2014 (test)			BEA-2019 (test)			JFLEG (test)
		P	R	F _{0.5}	P	R	F _{0.5}	GLEU
(Zhao et al., 2019)		67.7	40.6	59.8	–	–	–	–
(Awasthi et al., 2019)		66.1	43.0	59.7	–	–	–	60.3
(Kiyono et al., 2019)		67.9	44.1	61.3	65.5	59.4	64.2	59.7
(Kaneko et al., 2020)		69.2	45.6	62.6	67.1	60.1	65.6	61.3
(Lichtarge et al., 2019)	✓	66.7	43.9	60.4	–	–	–	63.3
(Zhao et al., 2019)	✓	71.6	38.7	61.2	–	–	–	61.0
(Awasthi et al., 2019)	✓	68.3	43.2	61.2	–	–	–	61.0
(Kiyono et al., 2019)	✓	72.4	46.1	65.0	74.7	56.7	70.2	61.4
(Kantor et al., 2019)	✓	–	–	–	78.3	58.0	73.2	–
(Kaneko et al., 2020)	✓	72.6	46.4	65.2	72.3	61.4	69.8	62.0
Baseline (BERT-base)		72.1	42.0	63.0	71.5	55.7	67.6	60.1
+GST		72.6	42.5	63.6	71.9	55.9	68.0	60.5
Baseline (RoBERTa-base)		73.9	41.5	64.0	77.2	55.1	71.5	60.6
+GST		74.1	42.2	64.4	77.5	55.7	71.9	60.9
Baseline (XLNet-base)		77.5	40.1	65.3	79.2	53.9	72.4	61.5
+GST		78.4	39.9	65.7	79.4	54.5	72.8	61.8

Table 1: Comparison of GEC models. The baseline comes from the model released by (Omelianchuk et al., 2020).

samples. This trains the model with more samples and more varied errors and alleviates the exposure bias issue. Separate cross-entropy losses are calculated for the Grammatical Error Detector and Labeler, and we detail the whole algorithm for our training process in Algorithm 3.

3 Detailed Training Process

To synthesize new errors based on a genuine grammatical error distribution, we add a sampling module to a trained GED module. Specifically, we use Gumbel-softmax sampling, a simple and efficient way to draw samples z from a categorical distribution with class probabilities P_{GEL} using the Gumbel-Max trick (Gumbel, 1954; Maddison et al., 2014):

$$z = \text{one_hot}(\text{argmax}_j [g_j + \log[P_{\text{GEL}}^i]_j]) \quad (1)$$

where $g_1 \dots g_j$ are i.i.d samples drawn from $\text{Gumbel}(0, 1)^2$. We use the softmax function as a continuous, differentiable approximation to argmax :

$$[y_i]_k = \frac{\exp((\log([P_{\text{GEL}}^i]_k) + g_k)/\tau)}{\sum_{j=1}^{|C|} \exp((\log([P_{\text{GEL}}^i]_j) + g_j)/\tau)}, \quad (2)$$

where $|C|$ is the number of classes, τ is the softmax temperature. Altering γ and β allows us to synthesize input samples of different error rates.

²The $\text{Gumbel}(0, 1)$ distribution can be sampled using inverse transform sampling by drawing $u \sim \text{Uniform}(0, 1)$ and computing $g = -\log(-\log(u))$.

Sampling	CoNLL-2014 (test)		
	P	R	F _{0.5}
Random	74.3	40.2	63.5
GumbelSoftmax	78.4	39.9	65.7
Multinomial	78.1	39.9	65.5

Table 2: Comparing the effects of different sampling distributions.

4 Experiments

4.1 Setup

To isolate our GAN-like Sequence Labeling Training (GST) approach, we use the same model setting and training details as in (Omelianchuk et al., 2020). The training data includes PIE’s synthetic data (Awasthi et al., 2019), NUCLE (Dahlmeier et al., 2013), Lang-8 (Tajiri et al., 2012), FCE (Yannakoudakis et al., 2011), Cambridge Learner Corpus (the publicly available portion) (Nicholls, 2003), and WI+LOCNESS (Bryant et al., 2019). Our models are evaluated on the test sets of CoNLL-2014 (Ng et al., 2014), BEA-2019 (Bryant et al., 2019), and JFLEG (Napoles et al., 2017) with the official M^2 (Dahlmeier and Ng, 2012), ERRANT (Bryant et al., 2017), and GLEU (Napoles et al., 2015) scorers, respectively.

4.2 Results and Analysis

Our results on the three test datasets are listed in Table 1. Our baseline model achieves the best single model CoNLL-2014 $F_{0.5}$, BEA-2019 $F_{0.5}$, and JFLEG GLEU scores, showing that the baseline we use is very strong. The results on the three

Model	CoNLL-2014 (test)		
	P	R	F _{0.5}
<i>Baseline</i>	72.1	42.0	63.0
<i>Baseline + Extended Training</i>	73.5	40.1	63.0
<i>Intermediate Outputs</i>	73.2	39.8	62.7
<i>GST</i>	72.6	42.5	63.6

Table 3: Comparing GST training with additional baselines.

benchmarks are further improved using the GST approach, which demonstrates that the GST approach can effectively alleviate the exposure bias issue. With GST, we achieved new best results on the CoNLL-2014 test dataset, surpassing ensemble methods while only using a single model.

In order to illustrate the benefits of sampling using Gumbel-Softmax, we replaced it with random sampling and Multinomial. The comparison is shown in Table 2. Random sampling actually hampers performance, which shows that synthetic sentences not based on a genuine error distribution do not alleviate exposure bias. Both GumbelSoftmax and Multinomial, which use a genuine error distribution, improve the model, though GumbelSoftmax appears to be more suitable for sampling in sequence labeling modeling.

In Figure 2, we show how the performance changes with increasing rounds of GST training. In the first few rounds, due to the model’s re-adaptation to new errors, there was a drop in performance on the test datasets; however, as the number of training rounds increased, performance on the test set gradually improved and finally stabilized.

Intermediate Outputs and Longer Training

In this experiment, we explored using intermediate outputs from our iterative inference process as additional training outputs to highlight the impact of generating new erroneous sentence by sampling from the real error distribution with our GST approach. For this experiment, we use our baseline architecture. As seen in the results in Table 3, whereas GST leads to a 0.6 $F_{0.5}$ gain over the baseline, using intermediate training outputs paired with golden sentences for additional training actually leads to worse performance, yielding a 0.3 $F_{0.5}$ loss in comparison to the baseline.

To confirm that GST’s performance gain is not due to the added training time, we also train the baseline for a commensurate amount of additional steps but find that this does not have any effect on model performance. This experiment demonstrates

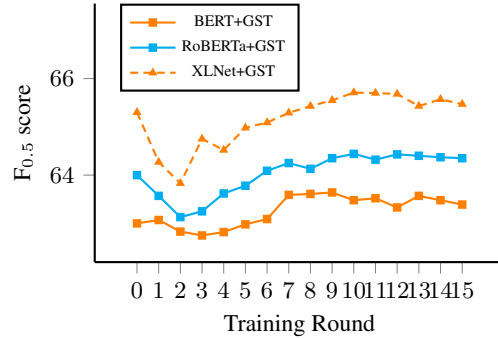


Figure 2: The GEC performance versus the the GST rounds on the CoNLL-2014 test set.

Model	CoNLL-2014 (test)		
	P	R	F _{0.5}
<i>Baseline</i>	72.1	42.0	63.0
<i>GST</i>	72.6	42.5	63.6
<i>Baseline w/o BERT</i>	65.0	32.6	54.2
<i>GST w/o BERT</i>	64.9	35.6	55.7

Table 4: Evaluating GST without pre-trained language models.

that our model does bring improvement to the baseline without relying on additional training steps. We also note that as our model is not significantly different in size from our baseline, our improvement is also not brought about by simply using a larger model.

Performance with out Pre-trained Language Models

We additionally explored the performance of our system in the absence of contextualized pre-trained language models. As we expected, these models make our model much more resilient to the exposure bias problem, and as seen in Table 4, the improvement brought about GST is therefore much more evident. In comparison to the baseline, using GST brings an improvement of 1.5 $F_{0.5}$ points.

5 Conclusion

In this paper, we studied the exposure bias problem GEC sequence labeling models face. To alleviate this issue, we proposed a novel GAN-like training method for the GEC sequence labeling model. Through evaluation on three GEC benchmarks, we demonstrate that our novel training approach further improves a strong baseline model, illustrating the effectiveness of our training approach. Notably, with the help of pre-trained language models and our training approach, we achieved state-of-the-art results on the CoNLL-2014 benchmark.

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.
- Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. [Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. [Learning to combine grammatical error corrections](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148, Florence, Italy. 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 (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. [Seq2seq dependency parsing](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020. [Data-dependent gaussian prior objective for language generation](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Zuchao Li, Zhuosheng Zhang, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2021. [Text compression-aided transformer encoding](#). *CoRR*, abs/2102.05951.
- Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. [Corpora generation for grammatical error correction](#). 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 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

- Xuezhe Ma and Eduard Hovy. 2016. [End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.
- Chris J. Maddison, Daniel Tarlow, and Tom Minka. 2014. [A* sampling](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3086–3094.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground truth for grammatical error correction metrics](#). 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 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanyskiy. 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. Association for Computational Linguistics.
- Kevin Parnow, Zuchao Li, and Hai Zhao. 2020. [Grammatical error correction: More data with more context](#). In *International Conference on Asian Language Processing, IALP 2020, Kuala Lumpur, Malaysia, December 4-6, 2020*, pages 24–29. IEEE.
- Vipul Raheja and Dimitris Alikaniotis. 2020. [Adversarial Grammatical Error Correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3075–3087, Online. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and aspect error correction for ESL learners using global context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.
- Zhuosheng Zhang, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Zuchao Li, and Hai Zhao. 2020a. [Neural machine translation with universal visual representation](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. [Semantics-aware BERT for language understanding](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.
- 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. Association for Computational Linguistics.