

# Pseudo-Bidirectional Decoding for Local Sequence Transduction

Wangchunshu Zhou<sup>1\*</sup> Tao Ge<sup>2</sup> Ke Xu<sup>1</sup>

<sup>1</sup>Beihang University, Beijing, China

<sup>2</sup>Microsoft Research Asia, Beijing, China

zhouwangchunshu@buaa.edu.cn, kexu@nlsde.buaa.edu.cn  
tage@microsoft.com

## Abstract

Local sequence transduction (LST) tasks are sequence transduction tasks where there exists massive overlapping between the source and target sequences, such as grammatical error correction and spell or OCR correction. Motivated by this characteristic of LST tasks, we propose Pseudo-Bidirectional Decoding (PBD), a simple but versatile approach for LST tasks. PBD copies the representation of source tokens to the decoder as pseudo future context that enables the decoder self-attention to attend to its bi-directional context. In addition, the bidirectional decoding scheme and the characteristic of LST tasks motivate us to share the encoder and the decoder of LST models. Our approach provides right-side context information for the decoder, reduces the number of parameters by half, and provides good regularization effects. Experimental results on several benchmark datasets show that our approach consistently improves the performance of standard seq2seq models on LST tasks.

## 1 Introduction

As illustrated in Figure 1, in local sequence transduction (LST) tasks, a model is trained to map an input sequence  $x_1, \dots, x_n$  to an output sequence  $y_1, \dots, y_m$ , where the input and output sequences are of similar length and differ only in a few positions. Many important NLP tasks can be formulated as LST tasks, including automatic grammatical error correction (GEC) (Lee and Seneff, 2006), OCR error correction (Tong and Evans, 1996) and spell checking (Fossati and Di Eugenio, 2007). With the recent success of sequence-to-sequence (seq2seq) learning (Sutskever et al., 2014) and the transformer model (Vaswani et al., 2017), most LST tasks have been tackled by directly training the transformer-based models in a seq2seq fashion.

\*This work was done during the first author’s internship at Microsoft Research Asia.

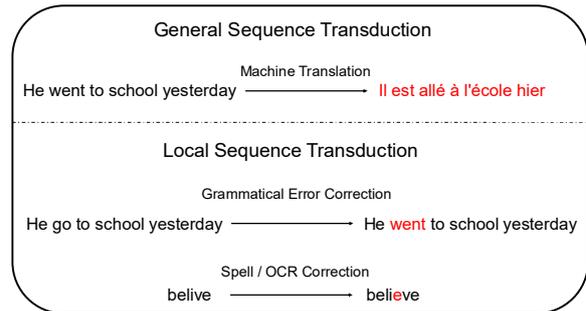


Figure 1: Illustration of the characteristic of local transduction tasks versus general sequence transduction tasks. Words and letters in red are those different from that in the input sequences.

While the conventional seq2seq paradigm suits well for general sequence transduction problems such as machine translation, their left-to-right autoregressive decoding scheme cannot access the future predictions on the right side, which does not fully utilize the characteristic of LST tasks and has been demonstrated to degrade the performance of seq2seq models (Zhang et al., 2018, 2019a).

In this work, motivated by the characteristic of LST tasks, we propose a Pseudo Bidirectional Decoding (PBD) approach to tackle LST tasks. Our approach copies the input tokens on the right side of the current decoding position as a proxy for the future tokens. In this way, we augment the decoder of the conventional transformer by allowing it to attend to the representation of “pseudo” future tokens in the decoder, making the decoding self-attention module bidirectional without introducing any computational overhead during inference. To retain the parallelizability of the training transformer models, we propose a novel masking strategy that enables the decoder to attend to copied future token representations during training in a parallelizable fashion. Also, we incorporate a segment embedding mechanism to make the decoder aware of whether

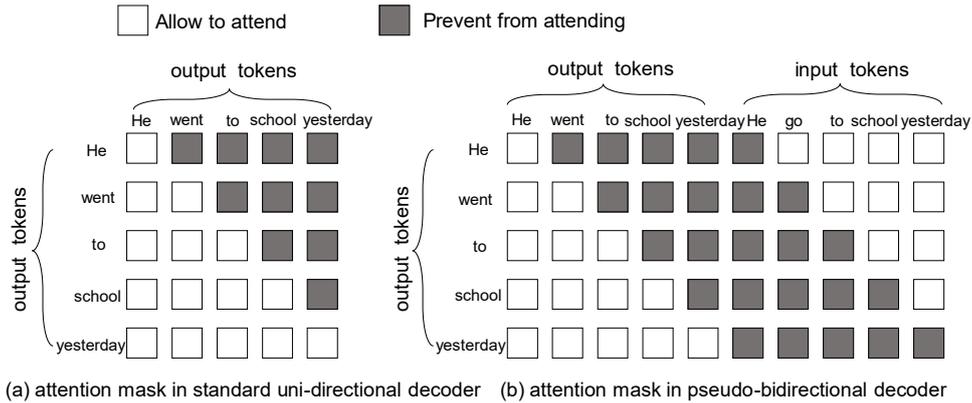


Figure 2: Illustration of the proposed pseudo future modeling approach and the pseudo-bidirectional attention mask used for parallel training.

a token is directly copied from the encoder and represent them differently from the generated tokens.

With the proposed approach, the encoder and the decoder in transformer models for LST tasks receive similar input sequences and both attend to their bidirectional context information, which motivates us to share all their parameters (except encoder-decoder attention). The parameter sharing mechanism allows us to roughly reduce the total number of parameters of the model by half, which is beneficial for real-world applications and makes the training more efficient. It also explicitly models the characteristic of LST tasks and leads to good regularization effects, enhancing the performance of transformer models on LST tasks and allowing us to train deeper models for further improvements.

We conduct extensive experiments on three LST tasks including grammatical error correction, spell correction, and OCR correction. Experimental results demonstrate that the proposed PBD approach is able to substantially and consistently improve over competitive transformer baselines across all three LST tasks and yield state-of-the-art results on both spell and OCR correction tasks.

## 2 Pseudo-Bidirectional Decoding

### 2.1 Pseudo Future Modeling

In contrast to recent works on bidirectional decoding (Zhang et al., 2018, 2019a) that employs a right-to-left model at the same time and combines their predictions in a post-hoc fashion with sophisticated algorithms, our approach enables the decoder of the seq2seq model to exploit the future context of the generated sequence without having to predict them in the first place.

Concretely, our method copies the representation of tokens from the  $N + 1$  th position to the end of the input sequence in the encoder layer to the corresponding decoder layer as pseudo future information when predicting the  $N$  th output token. For instance, for grammatical error correction, given an input text “*He go to school yesterday.*”, a conventional left-to-right decoder would probably generate “*goes*” at the second decoding step, as the decoder state is “*He \_*”, which is likely to be continued with the third person singular form of the verb “*go*”. In contrast, with the proposed pseudo-bidirectional decoding scheme, the decoder state becomes “*He \_ to school yesterday.*”, which facilitates the decoder to correctly generate “*went*”.

While ideally the encoder-decoder attention may capture this information, our method makes the decoder self-attention more effective by allowing it to directly attend to future information, which may be complementary to the information captured by the encoder-decoder attention module, leading to better empirical performance.

**Pseudo-bidirectional Attention Mask** A naive implementation of the PBD approach requires us to change the decoder input for each decoding step instead of feeding the entire output sequence into the decoder and use a causal attention mask to ensure the causality of the decoder. This would hinder the transformer model from being trained in parallel, thus makes the training much less efficient.

To address this problem, we propose a novel masking strategy. As illustrated in Figure 2, we concatenate the representation of the input sequences to that of the output sequences to form the key and the value in the decoder self-attention module. The pseudo-bidirectional attention mask makes the de-

coder self-attention bidirectional by allowing the query tokens to attend to pseudo future tokens copied from the encoder, retaining the causality of the decoder and enabling parallel training.

**Segment Embedding** While the characteristic of LST tasks ensures the copied pseudo future tokens to be similar to the expected output tokens, the simple position-wise alignment method may make the pseudo future information contain some noise. Therefore, we propose a simple segment embedding method that enables the decoder to distinguish the copied tokens from the tokens generated by the decoder and represent them differently.

Similar to BERT (Devlin et al., 2018), we add a learned embedding, which indicates whether the token is generated or copied, to each token representation in each decoder layer. Hopefully, this would make the decoder able to distinguish the copied tokens from the tokens generated by the decoder and represent them differently, thus improve its robustness to the noise in the copied future token representations.

## 2.2 Parameter sharing

The encoder and the decoder in conventional transformer models are independently parameterized for two main reasons. First, the inputs for the encoder and the decoder are usually different for general sequence transduction tasks such as machine translation and text summarization. Second, the encoder self-attention module is bidirectional whereas the decoder self-attention is causal (i.e. uni-directional).

The characteristic of LST tasks ensures the inputs for the encoder and the decoder to be roughly the same, and the proposed pseudo-bidirectional decoding method makes both the encoder and the decoder self-attention module to be bidirectional. This motivates us to share all parameters, except that in the encoder-decoder attention module, between the encoder and the decoder. This roughly reduces the number of parameters by half and also provide some regularization effects.

## 3 Experiments

In this section, we conduct experiments on LST benchmarks to validate the effectiveness of our approach. We mainly focus on the grammatical error correction task and also report results on two other LST tasks including spell and OCR corrections.

### 3.1 Grammatical Error Correction

**Datasets** Following the recent work (Grundkiewicz et al., 2019; Kiyono et al., 2019; Zhou et al., 2019a) in GEC, the GEC training data we use is the public Lang-8 (Mizumoto et al., 2011), NUCLE (Dahlmeier et al., 2013), FCE (Yanakoudakis et al., 2011) and W&I+LOCNESS datasets (Bryant et al., 2019; Granger, 1998). To investigate whether our approaches can yield consistent improvement in this setting, we pretrain our models with 30M sentence pairs obtained by the corruption-based approach and 30M pairs by the fluency boost back-translation approach (Ge et al., 2018) for GEC pre-training.

**Models** We use the “transformer-big” architecture as our baseline model, denoted by **Transformer**. For throughout comparison, we train two model variants with our approach. The first model (**Ours**) consists of the same number (i.e. 6) of transformer blocks with the baseline model, thus has the same inference latency while containing only half the number of parameters. The second model is denoted by **Ours-12 layers**, which consists of 12 transformer blocks, thus has approximately the same number of parameters but the inference latency is  $1.7\times$  longer. For comparison, we also train a variant of the “transformer-big” architecture with 12 blocks, which is denoted by **Transformer-12 layers**. For reference, we also compare with a recent model specifically designed for local sequence transduction tasks, denoted by **PIE**.

We use synthetic data for pre-training and then use the GEC training data to fine-tune the pre-trained models. The details of model training are provided in the Appendix due to space constraints.

**Evaluation** We evaluate the performance of GEC models on the BEA-19 and the CoNLL-14 benchmark datasets. Following the latest work in GEC (Grundkiewicz et al., 2019; Kiyono et al., 2019; Zhou et al., 2019a; Zhang et al., 2019b), we evaluate the performance of trained GEC models using  $F_{0.5}$  on test sets using official scripts<sup>1</sup> in both datasets.

**Results** The performance of different compared models on the GEC task is shown in Table 1. Note that we only compare against transformer models with the same pretraining/fine-tuning data in our setting for fair comparison as our contribution is

<sup>1</sup>M2scorer for CoNLL-14; Errant for BEA-19.

Method	# Parameters	BEA-19	CoNLL-14
PIE (with pretraining)	345M	-	59.7
w/o Pretraining			
Transformer	210M	57.1	51.5
Transformer-12 layers	383M	56.3	51.3
Ours	132M	58.6	53.7
-w/o future modeling		57.6	51.8
-w/o parameter sharing		58.2	52.9
Ours-12 layers	232M	<b>59.5</b>	<b>54.4</b>
-w/o future modeling		58.6	52.1
-w/o parameter sharing		58.8	53.8
w/ pretraining			
Transformer	210M	61.2	57.1
Transformer-12 layers	383M	61.9	57.5
Ours	132M	63.2	58.9
-w/o future modeling		61.5	57.4
-w/o parameter sharing		61.7	57.9
Ours-12 layers	232M	<b>63.9</b>	<b>60.1</b>
-w/o future modeling		61.8	57.7
-w/o parameter sharing		62.1	58.2

Table 1: The performance of different compared models on two test sets of GEC task.

orthogonal to better data synthesis method.

We can see that for the same configuration of the “transformer-big” baseline, our approach outperforms the baseline by a large margin in both settings with and without pretraining with synthetic data. This suggests that our approach is able to improve the performance of transformer-based LST models while reducing the number of parameters by half. In addition, we can see that by doubling the number of transformer blocks, our model is able to yield substantial further improvement. In contrast, we can see that simply increasing the number of transformer blocks (i.e. **Transformers-12 layers**) fails to improve the performance. This implies that our approach may be able to facilitate the training of deeper transformer models by providing regularization effects.

We then conduct an ablation study where either the pseudo future context modeling approach or the parameter sharing mechanism is disabled to better understand their relative importance. The results are shown in Table 1. We can see that the proposed pseudo future modeling approach method is very important in both the default and the deeper configuration of our transformer-based models, demonstrating its effectiveness on LST tasks. We also find that the parameter sharing mechanism is more effective in deeper models. This suggests that the parameter sharing mechanism may provide strong regularization effects and make it easier to train deeper transformer models.

Method	Spell	OCR
LSTM-soft	46.3	79.9
LSTM-hard	52.2	58.4
Ribeiro et al. (2018)	54.1	81.8
PIE	67.0	87.6
2 Layers		
Transformer	67.6	84.5
Ours	<b>69.2</b>	<b>88.7</b>
4 Layers		
Transformer-4 layers	67.1	85.4
Ours-4 layers	<b>70.4</b>	<b>89.6</b>

Table 2: The performance (accuracy) of different compared models on the spell and OCR correction tasks.

### 3.2 More Sequence Transduction Tasks

Following previous work (Ribeiro et al., 2018; Awasthi et al., 2019), we demonstrate the effectiveness of the proposed approaches on two additional local sequence transduction tasks including spell and OCR correction. We employ a two-layer transformer and a four-layer transformer as the backbone model for comparison and evaluate the compared models on the twitter spell correction dataset and the Finnish OCR dataset described as follows:

**Spell correction** We use the twitter spell correction dataset (Aramaki, 2010) which consists of 39172 pairs of original and corrected words obtained from twitter. We use the same train-dev-valid split as (Ribeiro et al., 2018) (31172/4000/4000). We tokenize on characters and our vocabulary comprises the 26 lower cased letters of English.

**OCR correction** We use the Finnish OCR dataset<sup>3</sup> by (Silfverberg et al., 2016) comprising words extracted from Early Modern Finnish corpus of OCR processed newspaper text. We use the same train-dev-test splits as provided by (Silfverberg et al., 2016). We tokenize on characters and our vocabulary comprises all the characters seen in the training data.

**Results** The result is shown in Table 2. We can see that the proposed method is able to significantly outperform the vanilla transformer-based models, as well as the LSTM and sequence labeling based LST baselines in both settings where either the number of parameters in the model is the same or the inference latency is the same, which is consis-

tent with the result in the GEC task. Our deeper model variant yields the state-of-the-art results in both tasks. This demonstrates the effectiveness of the proposed approach and suggests that our approach is versatile for different LST tasks.

## 4 Related work

**Local Sequence Transduction** Ribeiro et al. (2018) proposed to formulate LST tasks as sequence labeling problems by first predicting insert slots in the input sequences using learned insertion patterns and then using a sequence labeling task to output tokens in the input sequences or a special “delete” token. Awasthi et al. (2019); Malmi et al. (2019) propose to predict output edit operations including word transformations and further improve the performance of sequence labeling based LST models. Their approaches require massive engineering efforts to design an appropriate set of word transformations, which makes it non-trivial to generalize to other LST tasks. Also, the sequence labeling formulation lacks the flexibility of seq2seq models because it can only make local edits, which is demonstrated by their inferior performance. More recently, Li et al. (2020) use BERT to perform local sequence transduction with the method proposed by Zhou et al. (2019b). However, their method mainly suits for the cases where the length of the output sentence is unchanged.

### **Bidirectional Decoding and Future Modeling**

Previous work (Sennrich et al., 2016; Deng et al., 2018) investigate using right-to-left models to re-rank the generated sentences. More recently, Xia et al. (2017) and Zheng et al. (2018) propose two-pass decoding to model the right-side information, while Zhang et al. (2018, 2019a) use bidirectional beam search algorithms to generate the output sequences. These approaches integrate the right side context indirectly and introduce substantial computational overhead during inference, which is undesirable for real-world applications.

**Parameter Sharing in Transformer** Several parameter sharing mechanisms have been explored in transformer-based models. ALBERT (Lan et al., 2019) shares all encoder layers to reduce the number of parameters in the pretrained language model. Xia et al. (2019) propose to share the encoder and the decoder in transformer-based machine translation models. The performance gain in their setting is relatively small, which may be due to the dis-

crepancy in the input sequences and the attention direction in the encoder and the decoder.

## 5 Conclusion

In this paper, motivated by the characteristic of local sequence transduction tasks, we propose pseudo-bidirectional decoding (PBD) to provide approximated future information for transformer-based LST models and share the parameters between the encoder and the decoder of LST models to provide regularization effects while reducing the number of parameters. Our experiments on three LST tasks shows that our approach is able to yield consistent improvements upon strong transformer baselines while significantly reducing the number of parameters in the model.

## Acknowledgments

We thank the anonymous reviewers for their valuable comments.

## References

- Eiji Aramaki. 2010. Typo corpus.
- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. *arXiv preprint arXiv:1910.02893*.
- Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *BEA*.
- 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 *BEA*.
- Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba’s neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation*, pages 368–376.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Davide Fossati and Barbara Di Eugenio. 2007. A mixed trigrams approach for context sensitive spell checking. In *International conference on intelligent text processing and computational linguistics*.
- Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *ACL*.

- Sylviane Granger. 1998. *The computer learner corpus: a versatile new source of data for SLA research*. na.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of BEA*, pages 252–263.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *EMNLP-IJCNLP*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- John Lee and Stephanie Seneff. 2006. Automatic grammar correction for second-language learners. In *ICSLP*.
- Yiyuan Li, Antonios Anastasopoulos, and Alan W Black. 2020. Towards minimal supervision bert-based grammar error correction. *arXiv preprint arXiv:2001.03521*.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- 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 *In IJCNLP*.
- Joana Ribeiro, Shashi Narayan, Shay B Cohen, and Xavier Carreras. 2018. Local string transduction as sequence labeling. In *Proceedings of ICCL*, pages 1360–1371.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.
- Miikka Silfverberg, Pekka Kauppinen, Krister Lindén, et al. 2016. Data-driven spelling correction using weighted finite-state methods. In *SIGFSM Workshop on Statistical NLP and Weighted Automata, ACL. ACL*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.
- Xiang Tong and David A Evans. 1996. A statistical approach to automatic ocr error correction in context. In *Fourth Workshop on Very Large Corpora*.
- 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 *NIPS*, pages 5998–6008.
- Yingce Xia, Tianyu He, Xu Tan, Fei Tian, Di He, and Tao Qin. 2019. Tied transformers: Neural machine translation with shared encoder and decoder. In *AAAI*.
- Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *ACL*.
- Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2019a. Synchronous bidirectional inference for neural sequence generation. *arXiv preprint arXiv:1902.08955*.
- Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. In *AAAI*.
- Yi Zhang, Tao Ge, Furu Wei, Ming Zhou, and Xu Sun. 2019b. Sequence-to-sequence pre-training with data augmentation for sentence rewriting. *arXiv preprint arXiv:1909.06002*.
- Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2018. Modeling past and future for neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:145–157.
- Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2019a. Improving grammatical error correction with machine translation pairs. *arXiv preprint arXiv:1911.02825*.
- Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019b. Bert-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373.