Secoco: Self-Correcting Encoding for Neural Machine Translation

Tao Wang[1], Chengqi Zhao[1], Mingxuan Wang[1], Lei Li[2]*, Hang Li[1], Deyi Xiong[3]†
[1]ByteDance AI Lab [2]University of California, Santa Barbara
[3]College of Intelligence and Computing, Tianjin University, Tianjin, China
{wangtao.960826,zhaochengqi.d,wangmingxuan.89,lihang.lh}@bytedance.com
lilei@cs.ucsb.edu dyxiong@tju.edu.cn

Abstract

This paper presents **Se**lf-**co**rrecting En**co**ding (Secoco), a framework that effectively deals with input noise for robust neural machine translation by introducing self-correcting predictors. Different from previous robust approaches, Secoco enables NMT to explicitly correct noisy inputs and delete specific errors simultaneously with the translation decoding process. Secoco is able to achieve significant improvements of 1.6 BLEU points over strong baselines on two real-world test sets and a benchmark WMT dataset with good interpretability. The code and dataset are publicly available at https://github.com/rgwt123/Secoco.

1 Introduction

Neural machine translation (NMT) has witnessed remarkable progress in recent years (Bahdanau et al., 2015; Vaswani et al., 2017). Most previous works show promising results on clean datasets, such as WMT News Translation Shared Tasks (Barrault et al., 2020). However, inputs in real-world scenarios are usually with a wide variety of noises, which poses a significant challenge to NMT.

In order to mitigate this issue, we propose to build a noise-tolerant NMT model with a **Se**lf-**co**rrecting En**co**ding (Secoco) framework that explicitly models the error-correcting process as a sequence of operations: deletion and insertion. Figure 1 demonstrates a simple correcting process that transforms a noisy sequence "abbd" into its correct sequence "abcd" via a deletion and inserting operation. In order to learn desirable operations for noise correction given noisy inputs, we propose a insertion predictor and deletion predictor that predict appropriate deletion and insertion operations respectively. The two predictors work alternatively step by step to collectively transform a noisy input sequence into a clean sequence.

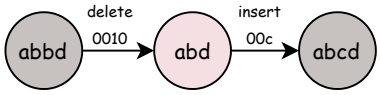

Figure 1: An example of the correcting and operation generation process. Assume we want to correct a synthesized noisy sequence “abbd" to its correct sequence “abcd". We can apply a deletion “b" operation to the third position (0010) and an insertion “c" operation to the third position (00c). (“abbd",“0010") and (“abd",“00c") can be regarded as training examples.

For training the two predictors, we collect a list of pairs (source sequence, operation sequence) (e.g., (“abbd",“0010") shown in Figure 1) from original training data by randomly deleting or inserting tokens from/to original clean sequences. With these collected training instances, we optimize the insertion and deletion predictors as well as NMT simultaneously in a multi-task learning way.

For inference, we propose two different variants for Secoco depending on the decoding modes. The first variant is an end-to-end approach like normal NMT decoding where the encoder is implicitly trained with self-correcting information. In this setting, we only predict operations during training and the encoder can have this kind of knowledge. The other variant is iterative editing, which corrects the input gradually and performs translation after the input is unchanged.

Compared with previous approaches, Secoco has two advantages. First, Secoco introduces a more explicit and direct way to model the noise correcting process. Second, Secoco enables an interpretable translation process. With the predicted operation sequence, it is easy to understand how the noisy input is corrected. We conduct experiments on three test sets, including Dialogue, Speech, and WMT14 En-De tasks. The results show that Secoco outper-

*Work is done while at ByteDance.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4639–4644
November 7–11, 2021.

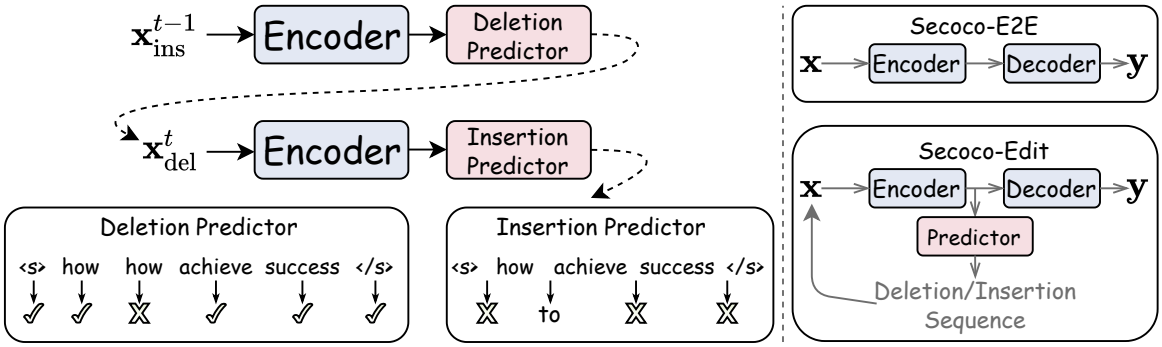

Figure 2: Diagram of the proposed Secoco. The left part is the illustration of self-correcting encoding. It contains a deletion predictor (Eq. 1) and an insertion predictor (Eq. 2). We omit the translation part here due to the space limit. The right part shows the decoding modes.

forms the baseline by $+1.6$ BLEU.

2 Approach

Our approach is illustrated in Figure 2. The left part of Figure 2 demonstrates the encoding module of Secoco. The only difference of Secoco from standard translation models is the two correcting operation predictors, which generate the operation sequence based on the encoder representation of an input text. The deletion predictor decides which word to be deleted while the insertion predictor decides which word to be inserted into which position. The combination of these two operations is able to simulate arbitrary complex correcting operations (Gu et al., 2019).

We illustrate the training data synthesizing process for the two predictors in Figure 1. It is worth noting that for correction that contains several iterations of editing (i.e., deletion or insertion), we sample only one iteration from it.

2.1 Self-Correcting Encoding

Secoco iteratively applies deletion and insertion operations to obtain a clean source sentence from a noisy input source sentence. Formally, given a source sentence $\mathbf{x}$, we introduce $\mathbf{x}^t_{\text{del}}$ and $\mathbf{x}^t_{\text{ins}}$ as the edited sentences at the t-th iteration after the deletion and insertion operation is respectively performed. As illustrated in the left part in Figure 2, the deletion predictor decides whether to delete (1) or keep unchanged (0) at position i:

$$p(c^t_i|\mathbf{x}^{t-1}_{\text{ins}}) = \text{sigmoid}(h^{t-1}_{\text{ins},i}\, W) \quad (1)$$

where $c^t_i \in \{0, 1\}$, $W \in \mathbb{R}^{d\times 2}$ and $h^{t-1}_{\text{ins},*} \in \mathbb{R}^{1\times d}$ is the encoded source representation after $(t-1)$ iterations.

Similarly, the insertion predictor considers the positions between each pair of neighboring words, and predicts a word to be inserted at position j:

$$p(w^t_j|\mathbf{x}^t_{\text{del}}) = \text{softmax}([h^t_{\text{del},j}; h^t_{\text{del},j+1}]\, Z) \quad (2)$$

where $Z \in \mathbb{R}^{2d\times(|V|+1)}$ and $h^t_{\text{del},*}$ is the encoded representation after deletion at the t-th iteration. Here, $|V|$ is the source vocabulary size and we append an empty token into the vocabulary, denoting no insertion operation at that position.

Although the iterative editing process relies heavily on previous operations for both the prediction of deletion and insertion, the two predictors and labels are independently trained for simplicity and the training of parameters are jointly done. The training data generated in advance is used to train both the deletion and insertion predictors simultaneously.

2.2 Training Objectives

We build the Secoco based on the encoder-decoder framework. Given a source sentence $\mathbf{x}$ and its target translation $\mathbf{y} = \{y_1, ..., y_m\}$, NMT directly models the conditional probability of the target sentence over the source sentence:

$$p\left(\mathbf{y}|\mathbf{x}\right) = \prod_{i=1}^{m} p\left(y_i|\mathbf{x}, y_{<i}\right) \quad (3)$$

As for deletion and insertion predictors, assume we have the supervision $\{\mathbf{c}^t, \mathbf{w}^t\}$ for each iteration $t \in 1, ..., T$. We can jointly train the above three tasks, and the training objective is to maximize the overall log-likelihood:

$$\log p(\mathbf{y}|\mathbf{x}) + \sum_{t=1}^{T} \left(\log p(\mathbf{c}^t|\mathbf{x}^{t-1}_{\text{ins}}) + \log p(\mathbf{w}^t|\mathbf{x}^t_{\text{del}})\right) \quad (4)$$

Test set	Size	Noise Types	Edits
Dialogue	1,931	dropped pronoun dropped punctuation typos	delete delete delete+insert
Speech	1,389	spoken words wrong punctuation	insert delete+insert
WMT	3,000	random insertion random deletion repeated words	insert delete insert

Table 1: Details of the three test sets.

where T is set to 1 when we only sample one iteration of editing during training.

2.3 Decoding Modes

During inference, we can either use the encoder-decoder model only (Secoco-E2E) or translate the edited sentence after iteratively applying deletion and insertion operations (Secoco-Edit), as illustrated in the right part of Figure 2.

In general, Secoco-E2E provides better robustness without sacrificing decoding speed. For Secoco-Edit, iterative editing enables better interpretability. Detailed editing operations provide a different perspective on how the model resists noise.

3 Experiments

3.1 Data

We conducted our experiments on three test sets, including Dialogue, Speech, and WMT14 En-De, to examine the effectiveness of Secoco.

Dialogue is a real-world Chinese-English dialogue test set constructed based on TV drama subtitles[1], which contains three types of natural noises (Wang et al., 2021). Speech is an in-house Chinese-English speech translation test set which contains various noise from ASR. To evaluate Secoco on different language pairs, we also used WMT14 En-De test sets to build a noisy test set with random deletion and insertion operations. Table 1 shows the details of the three test sets.

For Chinese-English translation, we used WMT2020 Chinese-English data[2] (48M) for Dialogue, and CCMT[3] (9M) for Speech. For WMT En-De, we adopted the widely-used WMT14 training data[4] (4.5M). We synthesized corresponding noisy data according to the noise types of the corresponding test set. The test sets and codes for synthesizing noisy data used in our experiments are available at https://github.com/rgwt123/Secoco.

3.2 Baselines

We compared our method against the following three baseline systems.

BASE One widely-used way to achieve NMT robustness is to mix raw clean data with noisy data to train NMT models. We refer to models trained with/without synthetic data as BASE/BASE+synthetic.

REPAIR To deal with noisy inputs, one might train a repair model to transform noisy inputs into clean inputs that a normally trained translation model can deal with. Both the repair and translation model are transformer-based models. As a pipeline model (repairing before translating), REPAIR may suffer from error propagation.

RECONSTRUCTION We follow Zhou et al. (2019) to develop a multi-task based method to solve the robustness problem. We construct triples (clean input, noisy input, target translation), and introduce an additional decoder to obtain clean inputs from noisy inputs. This method enables NMT to transform a noisy input into a clean input and pass this knowledge into the translation decoder.

3.3 Settings

In our studies, all translation models were Transformer-base. They were trained with a batch size of 32,000 tokens. The beam size was set to 5 during decoding. We used byte pair encoding compression algorithm (BPE) (Sennrich et al., 2016) to process all these data and restricted merge operations to a maximum of 30k separately. For evaluation, we used the standard Sacrebleu (Post, 2018) to calculate BLEU-4. All models were implemented based on Fairseq (Ott et al., 2019).

3.4 Results

Table 2 shows the translation results on Dialogue, Speech and WMT En-De. Clearly, all competitors substantially improve the baseline model in terms of BLEU. Secoco achieves the best performance on all three test sets, gaining improvements of 2.2, 0.7, and 0.4 BLEU-4 points over BASE+synthetic respectively. The improvements suggest the effectiveness of self-correcting encoding.

It is worth noting that the BLEU scores here are results on noisy test sets, so they are certainly lower

[1] https://github.com/rgwt123/DialogueMT
[2] http://www.statmt.org/wmt20/translation-task.html
[3] This corpus is a part of WMT2020.
[4] http://www.statmt.org/wmt14/translation-task.html

Methods	Dialogue BLEU	△	Speech BLEU	△	WMT En-De BLEU	△	AVG BLEU	△	Latency (ms/sent)
BASE	31.8	N/A	11.1	N/A	24.5	N/A	22.5	N/A	22
BASE +synthetic	32.6	+0.8	11.7	+0.6	24.8	+0.3	23.0	+0.5	21
REPAIR	33.2	+1.4	11.4	+0.3	25.0	+0.5	23.2	+0.7	36
RECONSTRUCTION	33.7	+1.9	11.8	+0.7	24.6	+0.1	23.4	+0.9	21
Secoco-Edit	34.1	+2.3	12.3	+1.2	**25.2**	**+0.7**	23.9	+1.4	24
Secoco-E2E	**34.8**	**+3.0**	**12.4**	**+1.3**	25.1	+0.6	**24.1**	**+1.6**	22

Table 2: Experiment results on the Dialogue, Speech and WMT En-De translation test set. We evaluate the average latency over the three test sets.

Iteration	Edition	Sentence
0		We has things to to do today
1	delete insert	We ~~has~~ things to ~~to~~ do today We have things to do today
2	no delete insert	 We have things to do today .
0		我不认识 只知道是个记者
1	no delete insert	 我不认识他 只知道他是个记者
0		要怪舅怪他父母
1	delete insert	要怪舅怪他父母 要怪就怪他父母

Table 3: Examples of the editing process using Secoco-Edit. Iteration 0 represents the raw sentence. ~~word~~ is to be deleted while word is to be inserted.

than the results without noise.

Among these test sets, Dialogue is much more noisy and informal than the other two test sets. Secoco-E2E achieves a BLEU score of 34.8, which is even 3 BLEU points higher than the baseline. Speech is very challenging and contains many errors introduced by ASR. The best BLEU score of Speech is only 12.4, achieved by Secoco-E2E. We have additional two interesting findings. First, the performance of Secoco-E2E and Secoco-Edit is very close. Therefore, it is better to use Secoco-E2E for its simplification and efficiency. Second, Secoco is more effective on the real-world test sets, showing its potential in real-world application.

3.5 Iterative Editing

As described in Section 2.3, we iteratively edit the input until the input is unchanged and then translate it. We present examples in Table 3. We can see that multiple deletions can be parallel, and the same is true for insertions. Because we try to make editing sequences as short as possible during the training process, we usually need only 1 to 3 iterations during inference. We get an average iteration number of 2.3 on our three test sets.

4 Related Work

Approaches to the robustness of NMT can be roughly divided into three categories. In the first research line, adversarial examples are generated with black- or white-box methods. The generated adversarial examples are then used to combine with original training data for adversarial training (Ebrahimi et al., 2018; Chaturvedi et al., 2019; Cheng et al., 2019; Michel et al., 2019; Zhao et al., 2018; Cheng et al., 2020).

In the second strand, a wide variety of methods have been proposed to deal with noise in training data (Schwenk, 2018; Guo et al., 2018; Xu and Koehn, 2017; Koehn et al., 2018; van der Wees et al., 2017; Wang and Neubig, 2019; Wang et al., 2018a,b, 2019).

Finally, efforts have been also explored to directly cope with naturally occurring noise in texts, which are closely related to our work. Heigold et al. (2018); Belinkov and Bisk (2018); Levy et al. (2019) focus on word spelling errors. Sperber et al.; Liu et al. (2019) study translation problems caused by speech recognition. Vaibhav et al. (2019) introduce back-translation to generate more natural synthetic data, and employ extra tags to distinguish synthetic data from raw data. Zhou et al. (2019) propose a reconstruction method based on one encoder and two decoders architecture to deal with natural noise for NMT. Different from ours, most of these works use the synthetic data in a coarse-grained and implicit way (i.e. simply combining the synthetic and raw data).

5 Conclusions

In this paper, we have presented a framework Secoco to build a noise-tolerant NMT model with self-correcting capability. With the proposed Secoco-

E2E and Secoco-Edit methods, Secoco exhibits both efficiency and interpretability. Experiments and analysis on the three test sets demonstrate that the proposed Secoco is able to improve the quality of NMT in translating noisy inputs, and make better use of synthetic data.

Acknowledgments

Deyi Xiong was partially supported by the National Key Research and Development Program of China (Grant No.2019QY1802) and Natural Science Foundation of Tianjin (Grant No.19JCZDJC31400). We would like to thank the three anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, et al. 2020. Proceedings of the fifth conference on machine translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Akshay Chaturvedi, Abijith KP, and Utpal Garain. 2019. Exploring the robustness of nmt systems to nonsensical inputs. *arXiv preprint arXiv:1908.01165*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. 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 11179–11189.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, et al. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 68–80.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739.

Omer Levy, Jacob Eisenstein, Marjan Ghazvininejad, et al. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. 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 3103–3114.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association*

for Computational Linguistics (Volume 2: Short Papers), pages 228–234.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthias Sperber, Jan Niehues, and Alex Waibel. Toward robust neural machine translation for noisy input sequences.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. 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 1916–1920.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018a. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304.

Tao Wang, Chengqi Zhao, Mingxuan Wang, Lei Li, and Deyi Xiong. 2021. Autocorrect in the process of translation—multi-task learning improves dialogue machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 105–112.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019. Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018b. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143.

Xinyi Wang and Graham Neubig. 2019. Target conditioned sampling: Optimizing data selection for multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5823–5828.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Shuyan Zhou, Xiangkai Zeng, Yingqi Zhou, Antonios Anastasopoulos, and Graham Neubig. 2019. Improving robustness of neural machine translation with multi-task learning. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 565–571.