Improving Non-autoregressive Neural Machine Translation with Monolingual Data

Jiawei Zhou
Harvard University
jzhou02@g.harvard.edu

Phillip Keung
Amazon Inc.
keung@amazon.com

Abstract

Non-autoregressive (NAR) neural machine translation is usually done via knowledge distillation from an autoregressive (AR) model. Under this framework, we leverage large monolingual corpora to improve the NAR model's performance, with the goal of transferring the AR model's generalization ability while preventing overfitting. On top of a strong NAR baseline, our experimental results on the WMT14 En-De and WMT16 En-Ro news translation tasks confirm that monolingual data augmentation consistently improves the performance of the NAR model to approach the teacher AR model's performance, yields comparable or better results than the best non-iterative NAR methods in the literature and helps reduce overfitting in the training process.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2014) has achieved impressive performance in recent years, but the autoregressive decoding process limits the translation speed and restricts low-latency applications. To mitigate this issue, many non-autoregressive (NAR) translation methods have been proposed, including latent space models (Gu et al., 2017; Ma et al., 2019; Shu et al., 2019), iterative refinement methods (Lee et al., 2018; Ghazvininejad et al., 2019), and alternative loss functions (Libovický and Helcl, 2018; Wang et al., 2019; Wei et al., 2019; Li et al., 2019; Shao et al., 2019). The decoding speedup for NAR models is typically 2-15× depending on the specific setup (e.g., the number of length candidates, number of latent samples, etc.), and NAR models can be tuned to achieve different trade-offs between time complexity and decoding quality (Gu et al., 2017; Wei et al., 2019; Ghazvininejad et al., 2019; Ma et al., 2019).

Although different in various aspects, all of these methods are based on transformer modules (Vaswani et al., 2017), and depend on a well-trained AR model to obtain its output translations to create targets for NAR model training. This training setup is well-suited to leverage external monolingual data, since the target side of the NAR training corpus is always generated by an AR model. Techniques like backtranslation (Sennrich et al., 2015a) are known to improve MT performance using monolingual data alone. However, to the best of our knowledge, monolingual data augmentation for NAR-MT has not been reported in the literature.

In typical NAR-MT model training, an AR teacher provides a consistent supervision signal for the NAR model; the source text that was used to train the teacher is decoded by the teacher to create synthetic target text. In this work, we use a large amount of source text from monolingual corpora to generate additional teacher outputs for NAR-MT training.

We use a transformer model with minor structural changes to perform NAR generation in a non-iterative way, which establishes stronger baselines than most of the previous methods. We demonstrate that generating additional training data with monolingual corpora consistently improves the translation quality of our baseline NAR system on the WMT14 En-De and WMT16 En-Ro translation tasks. Furthermore, our experiments show that NAR models trained with increasing amount of extra monolingual data are less prone to overfitting and generalize better on longer sentences.

In addition, we have obtained Ro→En and En→De results which are state-of-the-art for non-iterative NAR-MT, just by using more monolingual data.

	Parallel	En Mono.	Non-En Mono.
En-Ro	608,320	2,197,792	2,261,206
En-De	4,459,186	3,008,621	3,015,110

Table 1: Number of sentences per language arc. 'Mono' refers to the amount of monolingual text available.

2 Methodology

2.1 Basic Approach

Most of the previous methods treat the NAR modeling objective as a product of independent token probabilities (Gu et al., 2017), but we adopt a different point of view by simply treating the NAR model as a function approximator of an existing AR model.

Given an AR model and a source sentence, the translation process of the greedy output[1] of the AR model is a complex but deterministic function. Since the neural networks can be near-perfect nonlinear function approximators (Liang and Srikant, 2016), we can expect an NAR model to learn the AR translation process quite well, as long as the model has enough capacity. In particular, we first obtain the greedy output of a trained AR model, and use the resulting paired data to train the NAR model. Other papers on NAR-MT (Gu et al., 2017; Lee et al., 2018; Ghazvininejad et al., 2019) have used AR teacher models to generate training data, and this is a form of sequence-level knowledge distillation (Kim and Rush, 2016).

2.2 Model Structure

Throughout this paper, we focus on non-iterative NAR methods. We use standard transformer structures with a few small changes for NAR-MT, which we describe below.

For the target side input, most of the previous work simply copied the source side as the decoder's input. We propose a *soft copying method* by using a Gaussian kernel to smooth the encoded source sentence embeddings x^{enc}. Suppose the source and target lengths are T and T' respectively. Then the t-th input token for the decoder is $\sum_{i=1}^{T} x_i^{enc} \cdot K(i,t)$, where $K(i,t)$ is the Gaussian distribution evaluated at i with mean $\frac{T}{T'}t$ and variance σ^2. (σ^2 is a learned parameter.)

We modify the decoder self-attention mask so that it does not mask out the future tokens, and every token is dependent on both its preceding and succeeding tokens in every layer.

Gu et al. (2017), Lee et al. (2018), Li et al. (2019) and Wang et al. (2019) use an additional positional self-attention module in each of the decoder layers, but we do not apply such a layer. It did not provide a clear performance improvement in our experiments, and we wanted to reduce the number of deviations from the base transformer structure. Instead, we add positional embeddings at each decoder layer.

2.3 Length Prediction

We use a simple method to select the target length for NAR generation at test time (Wang et al., 2019; Li et al., 2019), where we set the target length to be $T' = T + C$, where C is a constant term estimated from the parallel data and T is the length of the source sentence. We then create a list of candidate target lengths ranging from $[T' - B, T' + B]$ where B is the half-width of the interval. For example, if $T = 5$, $C = 1$ and we used a half-width of $B = 2$, then we would generate NAR translations of length $[4, 5, 6, 7, 8]$, for a total of 5 candidates. These translation candidates would then be ranked by the AR teacher to select the one with the highest probability. This is referred to as length-parallel decoding in Wei et al. (2019).

3 NAR-MT with Monolingual Data

Augmenting the NAR training corpus with monolingual data provides some potential benefits. Firstly, we allow more data to be translated by the AR teacher, so the NAR model can see more of the AR translation outputs than in the original training data, which helps the NAR model generalize better. Secondly, there is much more monolingual data than parallel data, especially for low-resource languages.

Incorporating monolingual data for NAR-MT is straightforward in our setup. Given an AR model that we want to approximate, we obtain the source-side monolingual text and use the AR model to generate the targets that we can train our NAR model on.

4 Experimental Setup

Data We evaluate NAR-MT training on both the WMT16 En-Ro (around 610k sentence pairs) and the WMT14 En-De (around 4.5M sentence pairs) parallel corpora along with the associated WMT

[1] By 'greedy', we mean decoding with a beam width of 1.

Models	WMT16		WMT14	
	En→Ro	Ro→En	En→De	De→En
NAT-FT (Gu et al., 2017)	27.29	29.06	17.69	21.47
NAT-FT (+NPD s=10)	29.02	30.76	18.66	22.41
NAT-FT (+NPD s=100)	29.79	31.44	19.17	23.20
NAT-IR (i_{dec}=1) (Lee et al., 2018)	24.45	25.73	13.91	16.77
CTC (Libovický and Helcl, 2018)	19.93	24.71	17.68	19.80
imitate-NAT (Wei et al., 2019)	28.61	28.90	22.44	25.67
imitate-NAT (+LPD)	31.45	31.81	24.15	27.28
CMLM (Ghazvininejad et al., 2019)	27.32	28.20	18.05	21.83
FlowSeq (Ma et al., 2019)	29.73	30.72	23.72	28.39
FlowSeq (NPD n=30)	**32.20**	**32.84**	**25.31**	**30.68**
Our AR Transformer (beam 1)	33.56	33.68	28.84	32.77
Our AR Transformer (beam 4)	34.50	34.01	29.65	33.65
Our NAR baseline (B=5)	31.21	32.06	23.57	29.01
+ 50% monolingual data	31.74	33.16	25.35	29.78
+ monolingual data	31.91	33.46	25.53	29.96
+ monolingual data and de-dup	**31.96**	**33.57**	**25.73**	**30.18**
+ monolingual data and de-dup (doubled training time)†			**26.54**	**30.80**

Table 2: BLEU scores on the WMT16 En-Ro and WMT14 En-De test sets for different NAR models. All reported scores are from non-iterative NAR methods with similar hyper-parameter settings for transformers. 'de-dup' removes adjacent duplicated tokens. B is the half-width in Sec. 2.3. † marks results obtained by simply training the model for longer time until full convergence to fairly compare with previous works, with which we achieve further SOTA performance, but they are not directly comparable to our other experiments and are thus ignored in our discussion and analysis.

monolingual corpora for each language. For the parallel data, we use the processed data from Lee et al. (2018) to be consistent with previous publications. The WMT16 En-Ro task uses newsdev-2016 and newstest-2016 as development and test sets, and the WMT14 En-De task uses newstest-2013 and newstest-2014 as development and test sets. We report all results on test sets. We used the Romanian portion of the News Crawl 2015 corpus and the English portion of the Europarl v7/v8 corpus[2] as monolingual text for our En-Ro experiments, which are both about 4 times larger than the original paired data. We used the News Crawl 2007/2008 corpora for German and English monolingual text[2] in our En-De experiments, and downsampled them to ∼3 million sentences per language. The data statistics are summarized in Table 1. The monolingual data are processed following Lee et al. (2018), which are tokenized and segmented into subword units (Sennrich et al., 2015b). The vocabulary is shared between source and target languages and has ∼40k units. We use BLEU to evaluate the translation quality[3].

Model Configuration We use the settings for the base transformer configuration in Vaswani et al. (2017) for all the models: 6 layers per stack, 8 attention heads per layer, 512 model dimensions and 2048 hidden dimensions. The AR and NAR model have the same encoder-decoder structure, except for the decoder attention mask and the decoding input for the NAR model as described in Sec. 2.2.

Training and Inference We initialize the NAR embedding layer and encoder parameters with the AR model's. The NAR model is trained with the AR model's greedy outputs as targets. We use the Adam optimizer, with batches of size 64k tokens for one gradient update, and the learning rate schedule is the same as the one in Vaswani et al. (2017), where we use 4,000 warm-up steps and the

[2]http://www.statmt.org/wmt16/translation-task.html

[3]We report tokenized BLEU scores in line with prior work (Lee et al., 2018; Ma et al., 2019), which are case-insensitive for WMT16 En-Ro and case-sensitive for WMT14 En-De in the data provided by Lee et al. (2018).

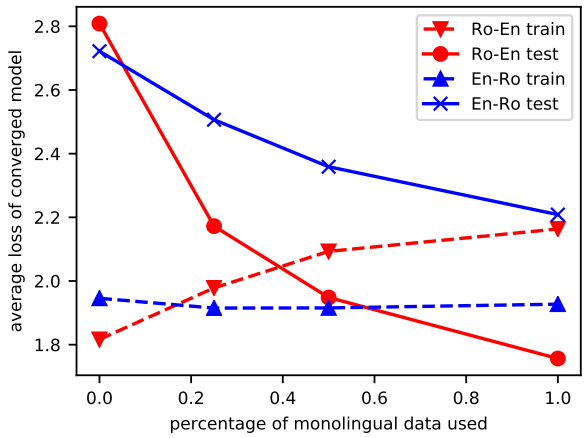

Figure 1: Average loss of the NAR models versus the percentage of monolingual data used during training. The test set losses decrease as more monolingual data is added, and the gap towards training losses are closing, which indicates that monolingual data augmentation reduces overfitting.

	En→Ro			Ro→En		
B	no mono	half mono	all mono	no mono	half mono	all mono
0	27.19	+0.65	+0.56	26.62	+1.52	+1.58
1	29.34	+0.63	+0.69	28.81	+1.26	+1.46
2	30.46	+0.34	+0.45	30.18	+1.08	+1.24
3	30.87	+0.37	+0.71	31.24	+0.88	+1.09
4	31.06	+0.45	+0.67	31.92	+0.90	+1.25
5	31.21	+0.53	+0.70	32.06	+1.10	+1.40
6	31.20	+0.39	+0.62	31.98	+1.17	+1.43
7	30.99	+0.43	+0.51	31.85	+1.19	+1.31
gold	29.64	+0.61	+0.85	29.83	+1.42	+1.69

Table 3: BLEU scores on the WMT16 En-Ro test sets for NAR models trained with different numbers of length candidates and amounts of additional monolingual data. The half-width B determines the number of length candidates (Sec. 2.3). 'gold' refers to using the true target length instead of predicting it. All the +deltas are relative to the 'no mono' case.

maximum learning rate is around 0.0014. We stop training when there is no further improvement in the last 5 epochs, and training finishes in 30 epochs for AR models and 50 epochs for NAR models, except for the En-De experiments with monolingual data where we train for 35 epochs to roughly match the number of parameter updating steps without using extra monolingual data (∼140k steps). We average the last 5 checkpoints to obtain the final model. We train the NAR model with cross-entropy loss and label smoothing ($\epsilon = 0.1$). During inference time, we use length parallel decoding with $C = 0$, and evaluate the BLEU scores on the reference sentences. All the models are implemented with MXNet and GluonNLP (Guo et al., 2019). We used 4 NVIDIA V100 GPUs for training, which takes about a day for an AR model and up to a week for an NAR model depending on the data size, and testing is performed on a single GPU.

5 Results and Analysis

Main Results We present our BLEU scores alongside the scores of other non-iterative methods in Table 2. Our baseline results surpass many of the previous results, which we attribute to the way that we initialize the decoding process. Instead of directly copying the source embeddings to the decoder input, we use an interpolated version of the encoder outputs as the decoder input, which allows the encoder to transform the source embeddings into a more usable form. Note that a similar technique is adopted in Wei et al. (2019), but our model structure and optimization are much simpler as we do not have any imitation module for detailed teacher guidance.

Our results confirm that the use of monolingual data improves the NAR model's performance, and the gain is proportional to the amount of extra monolingual data as seen from the BLUE scores with using only half and the full monolingual data. By incorporating all of the monolingual data for the En-Ro NAR-MT task, we see a gain of 0.70 BLEU points for the En→Ro direction and 1.40 for the Ro→En direction. Similarly, we also see significant gains in the En-De NAR-MT task, with an increase of 1.96 BLEU points for the En→De direction and 0.95 for the De→En direction.

By removing the duplicated output tokens as a simple postprocessing step (following Lee et al. (2018)), we achieved 33.57 BLEU for the WMT16 Ro→En direction and 25.73 BLEU for the WMT14 En→De direction, which are state-of-the-art among non-iterative NAR-MT results. In addition, our work shrinks the gap between the AR teacher and the NAR model to just 0.11 BLEU points in the Ro→En direction.

Losses in Training and Evaluation To further investigate how much the monolingual data contributes to BLEU improvements, we train En-Ro NAR models with 0%, 25%, 50%, and 100% of the monolingual corpora and plot the cross-entropy

loss on the training data and the testing data for the converged model. In Figure 1, when no monolingual data is used, the training loss typically converges to a lower point compared to the loss on the testing set, which is not the case for the AR model where the validation and testing losses are usually lower than the training loss. This indicates that the NAR model overfits to the training data, which hinders its generalization ability. However, as more monolingual data is added to the training recipe, the overfitting problem is reduced and the gap between the evaluation and training losses shrinks.

Effect of Length-Parallel Decoding To test how the NAR model performance and the monolingual gains are affected by the number of decoding length candidates, we vary the half-width B (Sec. 2.3) across a range of values and test the NAR models trained with 0%, 50%, and 100% of the monolingual data for the En-Ro task (Table 3). The table shows that having multiple length candidates can increase the BLEU score significantly and can be better than using the gold target length, but having too many length candidates can hurt the performance and slow down decoding (in our case, the optimal B is 5). Nonetheless, for every value of B, the BLEU score consistently increases when monolingual data is used, and more data brings greater gains.

BLEU under Different Sentence Lengths In Table 4, we present the BLEU scores on WMT16 Ro→En test sentences grouped by source sentence lengths. We can see that the baseline NAR model's performance drops quickly as sentence length increases, whereas the NAR model trained with monolingual data degrades less over longer sentences, which demonstrates that external monolingual data improves the NAR model's generalization ability.

6 Discussion

We found that monolingual data augmentation reduces overfitting and improves the translation quality of NAR-MT models. We note that the monolingual corpora are derived from domains which may be different from those of the parallel training data or evaluation sets, and a mismatch can affect NAR translation performance. Other work in NMT has examined this issue in the context of backtranslation (e.g., Edunov et al. (2018)), and we expect the conclusions to be similar in the NAR-MT case.

src length	# sent.	AR beam 1	NAR baseline	+half mono	+all mono
$[1, 20]$	865	32.12	29.96	30.94	31.10
$[21, 40]$	867	33.82	30.77	31.92	31.96
$[41, 60]$	228	35.13	29.59	31.33	31.81
$[61, 80]$	29	35.09	26.69	27.99	30.47
$[81, 120]$	8	34.13	16.47	28.92	29.47
$[121, 140]$	2	6.70	3.11	3.56	5.99

Table 4: BLEU scores for source sentences in different length intervals on the WMT16 Ro→En test set. The gold target length is provided during decoding.

There are several open questions to investigate: Are the benefits of monolingual data orthogonal to other techniques like iterative refinement? Can the NAR model perfectly recover the AR model's performance with much larger monolingual datasets? Are the observed improvements language-dependent? We will consider these research directions in future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language 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 6114–6123.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Jian Guo, He He, Tong He, Leonard Lausen, Mu Li, Haibin Lin, Xingjian Shi, Chenguang Wang, Junyuan Xie, Sheng Zha, Aston Zhang, Hang Zhang, Zhi Zhang, Zhongyue Zhang, and Shuai Zheng. 2019. Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing. *arXiv preprint arXiv:1907.04433*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. *arXiv preprint arXiv:1909.06708*.

Shiyu Liang and Rayadurgam Srikant. 2016. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*.

Jindřich Libovickỳ and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv preprint arXiv:1811.04719*.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. Flowseq: Non-autoregressive conditional sequence generation with generative flow. *arXiv preprint arXiv:1909.02480*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2019. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. *arXiv preprint arXiv:1911.09320*.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2019. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. *arXiv preprint arXiv:1908.07181*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

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.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. *arXiv preprint arXiv:1902.10245*.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1906.02041*.