

A Study of Non-autoregressive Model for Sequence Generation

Yi Ren*

Zhejiang University
rayeren@zju.edu.cn

Jinglin Liu*

Zhejiang University
jinglinliu@zju.edu.cn

Xu Tan

Microsoft Research Asia
xuta@microsoft.com

Zhou Zhao[†]

Zhejiang University
zhaozhou@zju.edu.cn

Sheng Zhao

Microsoft STC Asia
Sheng.Zhao@microsoft.com

Tie-Yan Liu

Microsoft Research Asia
tyliu@microsoft.com

Abstract

Non-autoregressive (NAR) models generate all the tokens of a sequence in parallel, resulting in faster generation speed compared to their autoregressive (AR) counterparts but at the cost of lower accuracy. Different techniques including knowledge distillation and source-target alignment have been proposed to bridge the gap between AR and NAR models in various tasks such as neural machine translation (NMT), automatic speech recognition (ASR), and text to speech (TTS). With the help of those techniques, NAR models can catch up with the accuracy of AR models in some tasks but not in some others. In this work, we conduct a study to understand the difficulty of NAR sequence generation and try to answer: (1) Why NAR models can catch up with AR models in some tasks but not all? (2) Why techniques like knowledge distillation and source-target alignment can help NAR models. Since the main difference between AR and NAR models is that NAR models do not use dependency among target tokens while AR models do, intuitively the difficulty of NAR sequence generation heavily depends on the strongness of dependency among target tokens. To quantify such dependency, we propose an analysis model called CoMMA to characterize the difficulty of different NAR sequence generation tasks. We have several interesting findings: 1) Among the NMT, ASR and TTS tasks, ASR has the most target-token dependency while TTS has the least. 2) Knowledge distillation reduces the target-token dependency in target sequence and thus improves the accuracy of NAR models. 3) Source-target alignment constraint encourages dependency

of a target token on source tokens and thus eases the training of NAR models.

1 Introduction

Non-autoregressive (NAR) models (Oord et al., 2017; Gu et al., 2017; Chen et al., 2019; Ren et al., 2019), which generate all the tokens in a target sequence in parallel and can speed up inference, are widely explored in natural language and speech processing tasks such as neural machine translation (NMT) (Gu et al., 2017; Lee et al., 2018; Guo et al., 2019a; Wang et al., 2019; Li et al., 2019b; Guo et al., 2019b), automatic speech recognition (ASR) (Chen et al., 2019) and text to speech (TTS) synthesis (Oord et al., 2017; Ren et al., 2019). However, NAR models usually lead to lower accuracy than their autoregressive (AR) counterparts since the inner dependencies among the target tokens are explicitly removed.

Several techniques have been proposed to alleviate the accuracy degradation, including 1) knowledge distillation (Oord et al., 2017; Gu et al., 2017; Guo et al., 2019a,b; Ren et al., 2019), 2) imposing source-target alignment constraint with fertility (Gu et al., 2017), word mapping (Guo et al., 2019a), attention distillation (Li et al., 2019b) and duration prediction (Ren et al., 2019). With the help of those techniques, it is observed that NAR models can match the accuracy of AR models for some tasks (Ren et al., 2019), but the gap still exists for some other tasks (Gu et al., 2017; Chen et al., 2019). Therefore, several questions come out naturally: (1) Why the gap still exists for some tasks? Are some tasks more difficult for NAR generation than others? (2) Why the techniques like knowledge distillation and source-target alignment can help NAR generation?

* Equal contribution.

[†] Corresponding author

The main difference between AR and NAR models is that NAR models do not consider the dependency among target tokens, which is also the root cause of accuracy drop of NAR models. Thus, to better understand NAR sequence generation and answer the above questions, we need to characterize and quantify the target-token dependency, which turns out to be non-trivial since the sequences could be of different modalities (i.e., speech or text). For this purpose, we design a novel model called COnditional Masked prediction model with Mix-Attention (CoMMA), inspired by the mix-attention in He et al. (2018) and the masked language modeling in Devlin et al. (2018): in CoMMA, (1) the prediction of one target token can attend to all the source and target tokens with mix-attention, and 2) target tokens are randomly masked with varying probabilities. CoMMA can help us to measure target-token dependency using the ratio of the attention weights on target context over that on full (both source and target) context when predicting a target token: bigger ratio, larger dependency among target tokens.

We conduct a comprehensive study in this work and obtain several interesting discoveries that can answer previous questions. First, we find that the rank of the target-token dependency among the three tasks is ASR>NMT>TTS: ASR has the largest dependency while TTS has the smallest. This finding is consistent with the accuracy gap between AR and NAR models and demonstrates the difficulty of NAR generation across tasks. Second, we replace the target sequence of original training data with the sequence generated by an AR model (i.e., through knowledge distillation) and use the new data to train CoMMA; we find that the target-token dependency is reduced. Smaller target-token dependency makes NAR training easier and thus improves the accuracy. Third, source-target alignment constraint such as explicit duration prediction (Ren et al., 2019) or implicit attention distillation (Li et al., 2019b) also reduces the target-token dependency, thus helping the training of NAR models.

The main contributions of this work are as follows:

- We design a novel model, conditional masked prediction model with mix-attention (CoMMA), to measure the token dependency for sequence generation.
- With CoMMA, we find that: 1) Among the

three tasks, ASR is the most difficult and TTS is the least for NAR generation; 2) both knowledge distillation and imposing source-target alignment constraint reduce the target-token dependency, and thus reduce the difficulty of training NAR models.

2 CoMMA

In this section, we analyze the token dependency in the target sequence with a novel conditional masked prediction model with mix-attention (CoMMA). We first introduce the design and structure of CoMMA, and then describe how to measure the target token dependency based on CoMMA.

2.1 The Design of CoMMA

It is non-trivial to directly measure and compare the target token dependency in different modalities (i.e., speech or text) and different conditional source modalities (i.e., speech or text). Therefore, we have several considerations in the design of CoMMA: 1) We use masked language modeling in BERT (Devlin et al., 2018) with source condition to train CoMMA, which can help measure the dependency on target context when predicting the current masked token. 2) In order to ensure the dependency on source and target tokens can be comparable, we use mix-attention (He et al., 2018) to calculate the attention weights on both source and target tokens in a single softmax function.

The model architecture of CoMMA is shown in Figure 1. Specifically, CoMMA differs from standard Transformer (Vaswani et al., 2017) as follows: 1) Some tokens are randomly replaced by a special mask token $\langle M \rangle$ with probability p , and the model is trained to predict original unmasked tokens. 2) We employ mix-attention mechanism (He et al., 2018) where layer i in the decoder can attend to itself and the layer i in the encoder at the same time and compute the attention weights in a single softmax function. We share the parameters of attention and feed-forward layer between the encoder and decoder. 3) Following He et al. (2018), we add source/target embedding to tell the model whether a token is from the source or target sequence, and also add position embedding with the positions of source and target tokens both starting from zero. 4) The encoder and decoder pre-net (Shen et al., 2018) vary in different tasks: For TTS, encoder pre-net consists of only embedding lookup table, and decoder pre-net consists of 2-layer dense network

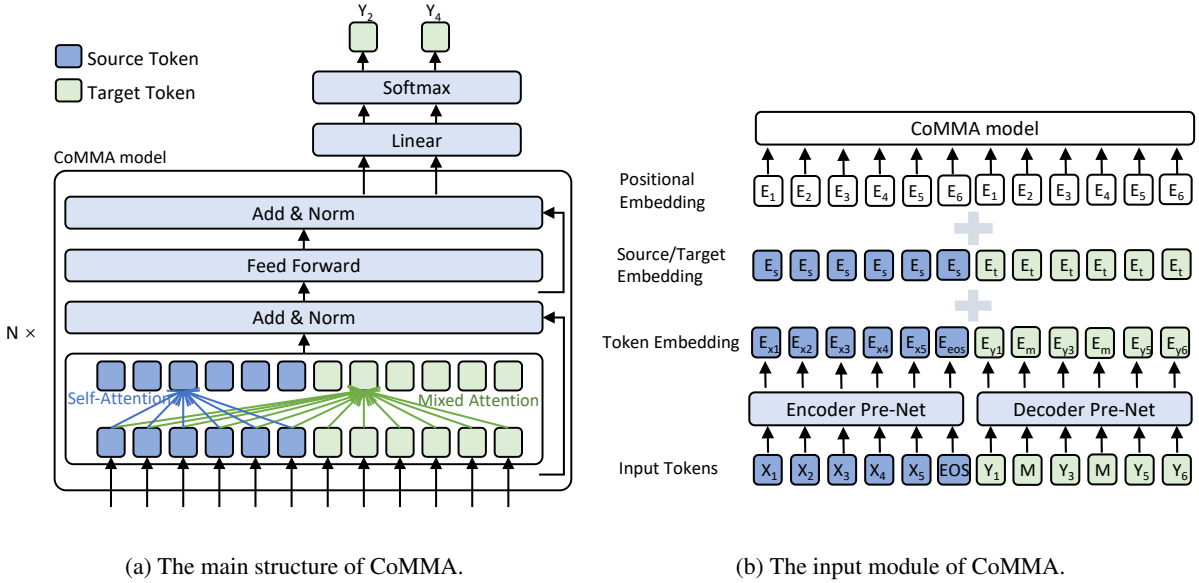


Figure 1: The architecture of conditional masked prediction model with mix-attention (CoMMA).

with ReLU activation. For ASR, encoder pre-net consists of 3-layer 2D convolutional network, and decoder pre-net consists of only embedding lookup table. For NMT, both encoder and decoder pre-net consist of only embedding lookup table.

CoMMA is designed to measure the target token dependency in a variety of sequence generations, including AR (unidirectional) generation, NAR generation, bidirectional generation or even identity copy. To this end, we vary the mask probability p (the ratio of the masked tokens in the whole target tokens¹) in a uniform distribution $p \sim U(0.0, 1.0)$ when training CoMMA. In this way, $p = 1$ covers NAR generation, $p = 0$ covers identity copy, and in some cases, p can also cover AR generation.

2.2 How to Measure Target Token Dependency based on CoMMA

To measure the target token dependency, we define a metric called attention density ratio R , which represents the ratio of the attention density (the normalized attention weights) on target context in mix-attention when predicting the target token with a well-trained CoMMA. We describe the calculation of R in the following steps.

First, we define the attention density ratio α for a single target token i as

$$\alpha_i = \frac{\frac{1}{N} \sum_{j=1}^N A_{i,j}}{\frac{1}{N} \sum_{j=1}^N A_{i,j} + \frac{1}{M} \sum_{j=N+1}^{N+M} A_{i,j}}, \quad (1)$$

¹Considering the continuity of the mel-spectrogram frames in speech sequence, we mask the frames by chunk, each chunk with frame size 10.

where $A_{i,j}$ denotes the attention weights from token i to token j in mix-attention, and $i \in [1, N]$ represents the target token while $j \in [N+1, N+M]$ represents the source token, M and N is the length of source and target sequence respectively, $\sum_{j=1}^{N+M} A_{i,j} = 1$. α_i represents the ratio of attention density on target context when predicting target token i .

Second, we average the attention density ratio α_i over all the predicted tokens (with masked probability p) in a sentence and get

$$\frac{1}{|\mathcal{M}^p|} \sum_{i \in \mathcal{M}^p} \alpha_i, \quad (2)$$

where \mathcal{M}^p represents the set of masked target tokens under mask probability p and $|\mathcal{M}^p|$ denotes the number of tokens in the set.

Third, for a given p , we calculate $R(p)$ over all test data and average them to get the final attention density ratio

$$R(p) = \text{Avg} \left(\frac{1}{|\mathcal{M}^p|} \sum_{i \in \mathcal{M}^p} \alpha_i \right). \quad (3)$$

We vary p and calculate $R(p)$ to measure the density ratio under different conditions, where a small p represents more target context that can be leveraged and a large p represents less context. In the extreme cases, $p = 1$ represent NAR generation while $p = 0$ represents to learn identity copy.

Given the proposed attention density ratio $R(p)$ based on CoMMA, we can measure the target token dependency of the NAR model in different tasks,

Task	NMT	ASR	TTS
AR	Transformer (Vaswani et al., 2017)	Transformer ASR (Karita et al., 2019)	Transformer TTS (Li et al., 2019a)
NAR	NAT (Gu et al., 2017) w/ AC	NAR-ASR (Chen et al., 2019) w/ AC	FastSpeech (Ren et al., 2019)

Table 1: The AR and NAR model we consider in each task. “AC” means attention constraint we mentioned in Section 5.

which can help understand a series of important research questions, as we introduce in the following three sections.

3 Study on the Difficulty of NAR Generation

In this section, we aim to find out why the gap still exists for ASR and NMT tasks, while in TTS, NAR can catch up with the accuracy of AR model. We also analyze the causes of different difficulties for different tasks. We start from evaluating the accuracy gap between AR and NAR models for NMT, ASR and TTS, and then measure the token dependency based on our proposed CoMMA.

3.1 The Accuracy Gap

We first train the AR and NAR models in each task and check the accuracy gap between AR and NAR models to measure the difficulty of NAR generation in each task.

Configuration of AR and NAR Model The AR and NAR models we considered are shown in Table 1, where we use Transformer as the AR models while the representative NAR models in each task. For a fair comparison, we make some modifications on the NAR models: 1) For ASR, we train a Transformer ASR first as teacher model and then constrain the attention distributions of NAR-ASR with the alignments converted by teacher attention weights, which will be introduced and discussed in Section 5. 2) For NMT, we constrain the KL-divergence of the encoder-to-decoder attention distributions between the AR and NAR models following Li et al. (2019b). We also list the hyperparameters of AR and NAR models for each task in Section A.

Datasets and Evaluations for NMT, ASR and TTS We conduct experiments on IWSLT 2014 German-English (De-En) translation dataset² for NMT, LibriTTS dataset (Zen et al., 2019) for ASR and LJSpeech dataset (Ito) for TTS. For

speech data, we transform the raw audio into mel-spectrograms following Shen et al. (2018) with 50 ms frame size and 12.5 ms hop size. For text data, we tokenize sentences with Moses tokenizer³ and then segment into subword symbols using Byte Pair Encoding (BPE) (Sennrich et al., 2015) for subword-level analysis, and convert the text sequence into phoneme sequence with grapheme-to-phoneme conversion (Sun et al., 2019) for phoneme-level analysis. We use BPE for NMT and ASR, while phoneme for TTS by default unless otherwise stated. We train all models on 2 NVIDIA 2080Ti GPUs using Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^{-9}$ and following the same learning rate schedule in (Vaswani et al., 2017).

For ASR, we evaluate word error rate (WER) on test-clean set in LibriTTS dataset. For NMT, we evaluate the BLEU score on IWSLT 2014 De-En test set. For TTS, we randomly split the LJSpeech dataset into 3 sets: 12500 samples for training, 300 samples for validation and 300 samples for testing, and then evaluate the mean opinion score (MOS) on the test set to measure the audio quality. The output mel-spectrograms of TTS model are transformed into audio samples using the pretrained WaveGlow (Prenger et al., 2019). Each audio is listened by at least 20 testers, who are all native English speakers.

Task	Model	Accuracy
NMT (BLEU/WER)	<i>Transformer</i>	33.90/47.18
	<i>NAT</i>	27.12/54.90
ASR (BLEU/WER)	<i>Transformer ASR</i>	66.60/20.10
	<i>NAR-ASR</i>	39.23/36.20
TTS (MOS)	<i>Transformer TTS</i>	3.82 ± 0.08
	<i>FastSpeech</i>	3.79 ± 0.12

Table 2: The accuracy gap between NAR and AR models.

Results of Accuracy Gap The accuracies of the AR and NAR models in each task are shown in

³<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl>

²<https://wit3.fbk.eu/mt.php?release=2014-01>

Table 2. It can be seen that NAR model can match the accuracy of AR model gap in TTS, while the gap still exists in ASR and NMT. We calculate both the WER and BLEU metrics in ASR and NMT for better comparison. It can be seen that ASR has a larger gap than NMT. Larger accuracy gap may indicate more difficult for NAR generation in this task. Next, we try to understand what factors influence difficulties among different tasks.

3.2 The Token Dependency

In the last subsection, we analyze the difficulty of NAR models from the perspective of the accuracy gap. In this subsection, we try to find evidence from the target token dependency, which is supposed to be consistent with the accuracy gap to measure the task difficulty.

Configuration of CoMMA We train CoMMA with the same configuration on NMT, ASR and TTS: the hidden size and the feed-forward hidden size and the number of layers are set to 512, 1024 and 6 respectively. We list other hyperparameters of CoMMA in Section B. We also use the same datasets for each task as described in Section 3.1 to train CoMMA.

Results of Token Dependency We use the attention density ratio calculated from CoMMA (as described in Section 2.2) to measure the target token dependency and show the results in Figure 2. It can be seen that the rank of attention density ratio $R(p)$ is $ASR > NMT > TTS$ for all p . Considering that $R(p)$ measures how much context information from target side is needed to generate a target token, we can see that ASR has more dependency on the target context and less on the source context, while TTS is the opposite, which is consistent with the accuracy gap between AR and NAR models as we described in Section 3.1.

As we vary p from 0.1 to 0.5, $R(p)$ decreases for all tasks since more tokens in the target side are masked. We also find that $R(p)$ in NMT decreases quicker than the other two tasks, which indicates that NMT is good at learning from source context when less context information can be leveraged from the target side while $R(p)$ in ASR decreases little. This can also explain why NAR in NMT achieves less gap than ASR.

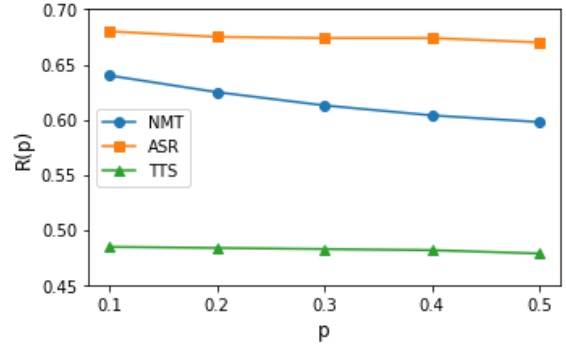


Figure 2: Attention density ratio $R(p)$ under different p in different tasks for performance gap analysis.

4 Study on Knowledge Distillation

In the current and next sections, we investigate why some techniques can help NAR generation from the aspect of target token dependency. We only analyze knowledge distillation and attention alignment techniques which are widely used in NAR, but we believe our analysis method can be applied to other NAR techniques, such as iterative refinement (Lee et al., 2018), fine-tuning from an AR model (Guo et al., 2019b) and so on.

Most existing NAR models (Oord et al., 2017; Gu et al., 2017; Wang et al., 2019; Guo et al., 2019a,b; Ren et al., 2019) rely on the technique of knowledge distillation, which generates the new target sequence given original source sequence from a pre-trained AR model and trains the NAR model for better accuracy. In this section, we first conduct experiments to verify the accuracy improvements of knowledge distillation. Next, based on our proposed CoMMA, we analyze why knowledge distillation could help NAR models.

4.1 The Effectiveness of Knowledge Distillation

Knowledge Distillation for NAR Models

Given a well-trained AR model θ_T and source sequence $x \in \mathcal{X}$ from the original training data, a new target sequence can be generated through

$$y' \sim P(y|x; \theta_T). \quad (4)$$

We can use beam search for NMT and ASR and greedy search for TTS to generate y' . Given the set of generated sequence pairs $(\mathcal{X}, \mathcal{Y}')$, we train the NAR models with negative log-likelihood loss

$$\mathcal{L}((\mathcal{X}, \mathcal{Y}'); \theta) = - \sum_{(x, y') \in (\mathcal{X}, \mathcal{Y}')} \log P(y'|x; \theta), \quad (5)$$

where θ is the parameters set of the NAR model.

Task	Model	Accuracy
NMT (BLEU)	<i>Transformer</i>	33.90
	<i>NAT</i>	27.12
	<i>NAT w/o KD</i>	21.79
TTS (MOS)	<i>Transformer TTS</i>	3.82 ± 0.08
	<i>FastSpeech</i>	3.79 ± 0.12
	<i>FastSpeech w/o KD</i>	3.58 ± 0.13

Table 3: The comparison between NAR models with and without knowledge distillation.

Experimental Results We only conducted knowledge distillation on NMT and TTS since there is no previous works on ASR yet. We train the NAR models in NMT and TTS with raw target token sequence instead of teacher outputs and compare the results with that in Table 2. The accuracy improvements of knowledge distillation are shown in Table 3. It can be seen that knowledge distillation can boost the accuracy of NAR in NMT and TTS, which is consistent with the previous works.

4.2 Why Knowledge Distillation Works

Recently, Zhou et al. (2019) find that knowledge distillation can reduce the complexity of data sets and help NAT to better model the variations in the output data. However, this explanation is reasonable on its own, but mainly from the perspective of data level and is not easy to understand. In this subsection, we analyze knowledge distillation from a more understandable and intuitive perspective, by observing the change of the token dependency based on our proposed CoMMA.

We measure the target token dependency by training CoMMA with the original training data and new data generated through knowledge distillation, respectively. The results are shown in Figure 3. It can be seen that knowledge distillation can decrease the attention density ratio $R(p)$ on both tasks, indicating that knowledge distillation can reduce the dependency on the target-side context when predicting a target token, which can be helpful for NAT model training.

5 Study on Alignment Constraint

Without the help of target context, NAR models usually suffer from ambiguous attention to the source context, which affects the accuracy. Re-

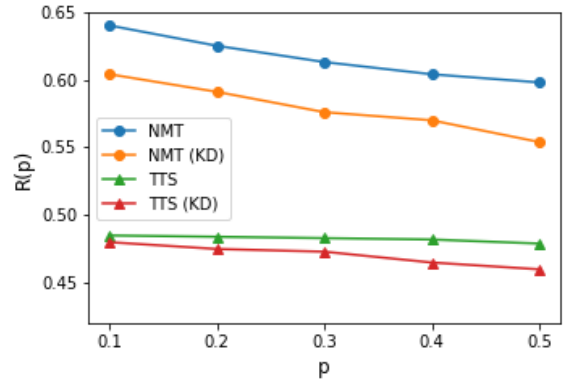


Figure 3: Attention density ratio $R(p)$ for NMT and TTS tasks under different p with and without knowledge distillation, where “KD” means knowledge distillation.

cently, many works have proposed a variety of approaches to help with the source-target alignment of NAR models, which can improve the estimation of the soft alignment in attention mechanism model. For example, Li et al. (2019b) constrain the KL-divergence of the encoder-to-decoder attention distributions between the AR and NAR models. Gu et al. (2017) predict the fertility of the source tokens to approximate the alignments between target sequence and source sequence. Guo et al. (2019a) convert the source token to target token with phrase table or embedding mapping for alignments. Ren et al. (2019) predict the duration (the number of mel-spectrograms) of each phoneme.

In this section, we first study the effectiveness of alignment constraint for NAR models, and then analyze why alignment constraint can help the NAR models by observing the changes of token dependency based on our proposed CoMMA.

5.1 The Effectiveness of Alignment Constraint

Alignment Constraint for NAR Models We choose the attention constraint mechanism which is commonly used based on previous works for each task.

For NMT, we follow Li et al. (2019b) to minimize the KL-divergence between the attention distributions of AR and NAR model as follow:

$$\mathcal{L}_{ac} = \frac{1}{N} \sum_{i=1}^N D_{KL}(A'_i || A_i), \quad (6)$$

where A'_i and A_i denote the source-target attention weights from the AR teacher model and NAR student model respectively. $A', A \in \mathbb{R}^{N \times M}$ where N

and M are the number of tokens in the target and source sequence.

For TTS, we follow Ren et al. (2019) to extract the encoder-to-decoder attention alignments from the well-trained AR teacher model and convert them to phoneme duration sequence, and then train the duration predictor to expand the hidden of the source sequence to match the length of target sequence.

For ASR, since there is no previous work proposing alignment constraint for NAR, we design a new alignment constraint method and explore its effectiveness. We first calculate the expectation position of teacher’s attention distributions for i -th target token: $E_i = \sum_{j=1}^M j * A'_{i,j}$ and cast it to the nearest integer. Then we constrain the attention weights of i -th target token for NAR model so that it can only attend to the source position between E_{i-1} and E_{i+1} . Specially, the first target token can only attend to the source position between 1 and E_2 while the last target token can only attend to the position between E_{N-1} and M . We apply this alignment constraint for ASR only in the training stage.

Task	Model	Accuracy
NMT (BLEU)	<i>Transformer</i>	33.90
	<i>NAT</i>	27.12
	<i>NAT w/o AC</i>	25.03
ASR (WER)	<i>Transformer ASR</i>	20.1
	<i>NAR-ASR</i>	33.1
	<i>NAR-ASR w/o AC</i>	39.23
TTS (MOS)	<i>Transformer TTS</i>	3.82 ± 0.08
	<i>FastSpeech</i>	3.79 ± 0.12
	<i>FastSpeech w/o AC</i>	1.97 ± 0.16

Table 4: The comparison between NAR models with and without alignment constraint.

Experimental Results We follow the model configuration and datasets as described in Section 3.1, and explore the accuracy improvements when adding attention constraint to NAR models. The results are shown in Table 4. It can be seen that attention constraint can not only improve the performance of NMT and TTS as previous works (Li et al., 2019b; Ren et al., 2019) demonstrated, but also help the NAR-ASR model achieve better scores.

5.2 Why Alignment Constraint Works

We further analyze how alignment constraint could help on NAR models by measuring the changes

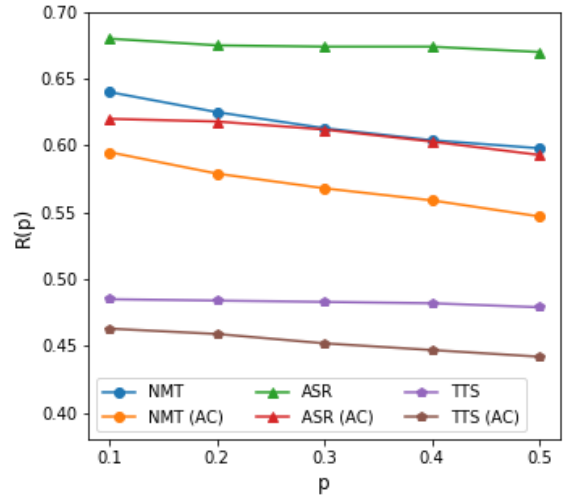


Figure 4: Attention density ratio $R(p)$ for NMT, ASR and TTS tasks under different p with and without alignment constraint (AC).

of token dependency when adding alignment constraint on CoMMA.

For simplicity, we use the method described in Equation 6 to help the training of CoMMA, where the teacher model is the AR model and student model is CoMMA. We minimize KL-divergence between the per-head encoder-to-decoder attention distributions of the AR model and CoMMA. First, we normalize the encoder-to-decoder attention weights in each head of mix-attention to convert each row of the attention weights to a distribution:

$$\hat{A}_{i,j} = \frac{A_{i,N+j}}{\sum_{k=1}^M A_{i,N+k}} \quad (7)$$

for each $i \in [1, N], j \in [1, M]$,

where $A \subset \mathbb{R}^{N \times (N+M)}$ is the weights of mix-attention described in Section 2.2, $\hat{A} \subset \mathbb{R}^{N \times M}$ is the normalized encoder-to-decoder attention weights, M and N is the length of source and target sequence. Then, we compute the KL-divergence loss for each head as follows:

$$\mathcal{L}_{ac} = \frac{1}{N} \sum_{i=1}^N D_{KL}(A'_i || \hat{A}_i), \quad (8)$$

where $A' \subset \mathbb{R}^{N \times M}$ is the encoder-to-decoder attention of AR teacher model. We average \mathcal{L}_{ac} over all heads and layers and get the final attention constraint loss for CoMMA.

We measure the token dependency by calculating the attention density ratio $R(p)$ based on CoMMA,

and show the results in Figure 4. It can be seen that alignment constraint can help reduce ratio $R(p)$ on each task and thus reduce the dependency on target context when predicting target tokens. In the meanwhile, alignment constraint can help the model extract more information from the source context, which can help the learning of NAR models.

Another interesting finding is that NAR model in TTS benefits from attention constraint most as shown in Table 4, and in the meanwhile, TTS has the least attention density ratio as shown in Figure 4. These observations suggest that NAR models with small target token dependency could benefit largely from alignment constraint.

6 Related Works

Several works try to analyze and understand NAR models on different tasks. We discuss these analyses from the two aspects: knowledge distillation and source-target alignment constraint.

Knowledge Distillation Knowledge distillation has long been used to compress the model size (Hinton et al., 2015; Furlanello et al., 2018; Yang et al., 2018; Anil et al., 2018; Li et al., 2017) or transfer the knowledge of teacher model to student model (Tan et al., 2019; Liu et al., 2019a,b), and soon been applied to NAR models (Gu et al., 2017; Oord et al., 2017; Guo et al., 2019a; Wang et al., 2019; Li et al., 2019b; Guo et al., 2019b; Ren et al., 2019) to boost the accuracy. Some works focus on studying why knowledge distillation works: Phuong and Lampert (2019) provide some insights into the mechanisms of knowledge distillation by studying the special case of linear and deep linear classifiers and find that data geometry, optimization bias and strong monotonicity determine the success of distillation; Yuan et al. (2019) argue that the success of KD is also due to the regularization of soft targets, which might be as important as the similarity information between categories.

However, few works have studied the cause of why knowledge distillation benefits NAR training. Recently, Zhou et al. (2019) investigate why knowledge distillation is important for the training of NAR model in NMT task and find that knowledge distillation can reduce the complexity of data sets and help NAR model to learn the variations in the output data.

Li et al. (2019b) explore the causes of the poor performance of the NAR model by observing the

attention distributions and hidden states of NAR model. Lee et al. (2018) presents some experiments and analysis to prove the necessity for multiple iterations generation for NAT. They also investigate the effectiveness of knowledge distillation in different task and make the assumption that teacher model can essentially clean the training data so that the distilled NAR model substantially outperforms NAR model trained with raw data.

Attention Alignment Constraint Previous work pointed out that adding additional alignment knowledge can improve the estimation of the soft alignment in attention mechanism model. For example, Chen et al. (2016) uses the Viterbi alignments of the IBM model 4 as an additional knowledge during NMT training by calculating the divergence between the attention weights and the statistical alignment information.

Compared with AR model, the attention distributions of NAR model are more ambiguous, which leads to the poor performance of the NAR model. Recent works employ attention alignment constraint between the well-trained AR and NAR model to train a better NAR model. Li et al. (2019b) leverages intermediate hidden information from a well-trained AR-NMT teacher model to improve the NAR-NMT model by minimizing KL-divergence between the per-head encoder-decoder attention of the teacher and the student. Ren et al. (2019) choose the encoder-decoder attention head from the AR-TTS teacher as the attention alignments to improve the performance of the NAR model in TTS.

7 Conclusion

In this paper, we conducted a comprehensive study on NAR models in NMT, ASR and TTS tasks to analyze several research questions, including the difficulty of NAR generation and why knowledge distillation and alignment constraint can help NAR models. We design a novel CoMMA and a metric called attention density ratio to measure the dependency on target context when predicting a target token, which can analyze these questions in a unified method. Through a series of empirical studies, we demonstrate that the difficulty of NAR generation correlates on the target token dependency, and knowledge distillation as well as alignment constraint reduces the dependency of target tokens and encourages the model to rely more on source context for target token prediction, which improves the

accuracy of NAR models. We believe our analyses can shed light on the understandings and further improvements on NAR models.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (Grant No.2018AAA0100603), Zhejiang Natural Science Foundation (LR19F020006), National Natural Science Foundation of China (Grant No.61836002), National Natural Science Foundation of China (Grant No.U1611461), and National Natural Science Foundation of China (Grant No.61751209). This work was also partially funded by Microsoft Research Asia. Thanks Tao Qin for the valuable suggestions, comments and guidance on this paper.

References

- Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. 2018. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*.
- Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak. 2019. Non-autoregressive transformer automatic speech recognition. *arXiv preprint arXiv:1911.04908*.
- Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. [Guided alignment training for topic-aware neural machine translation](#). *CoRR*, abs/1607.01628.
- 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*.
- Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770*.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.
- Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019a. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3723–3730.
- Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2019b. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1911.08717*.
- Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems*, pages 7944–7954.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Keith Ito. The lj speech dataset, 2017a. url <https://keithito.com/LJ-Speech-Dataset>.
- Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyang Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on transformer vs rnn in speech applications. *arXiv preprint arXiv:1909.06317*.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.
- Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and M Zhou. 2019a. Neural speech synthesis with transformer network. AAAI.
- Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936.
- Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019b. Hint-based training for non-autoregressive machine translation. *arXiv preprint arXiv:1909.06708*.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*.
- Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019b. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*.
- Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. 2017. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.
- Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151.
- Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics*,

- Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.
- Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. FastSpeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.
- Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-level ensemble distillation for grapheme-to-phoneme conversion. In *INTERSPEECH*.
- Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*.
- 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. In *AAAI*.
- Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan Yuille. 2018. Knowledge distillation in generations: More tolerant teachers educate better students. *arXiv preprint arXiv:1805.05551*.
- Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2019. Revisit knowledge distillation: a teacher-free framework. *arXiv preprint arXiv:1909.11723*.
- Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.
- Chunting Zhou, Graham Neubig, and Jiatao Gu. 2019. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727*.

A Model Settings of NAR and AR

We show the model settings of NAR and AR in Table 5. The hyperparameters in pre-net follow the methods in each task listed in Table 1 in the main part of the paper.

Transformer Hyperparameter	NMT / NAT	ASR / NAR-ASR	TTS / FastSpeech
Embedding Dimension	512	512	512
Encoder Layers	6	6	6
Encoder Hidden	512	512	512
Encoder Filter Size	1024	1024	1024
Encoder Heads	4	4	4
Decoder Layers	6	6	6
Decoder Hidden Size	512	512	512
Decoder Filter Size	1024	1024	1024
Decoder Heads	4	4	4
Dropout	0.2	0.1	0.2
Batch Size	64	32	32
Base Learning Rate	1e-3	1e-3	1e-3

Table 5: Hyperparameters of transformer-based AR and NAR models.

B Model Settings of CoMMA

We show the model settings of CoMMA in Table 6.

Name	Hyperparameter
Embedding Dimension	512
Encoder Layers	6
Encoder Hidden	512
Encoder Filter Size	1024
Encoder Heads	4
Decoder Layers	6
Decoder Hidden Size	512
Decoder Filter Size	1024
Decoder Heads	4
Dropout	0.1
Batch Size	64
Base Learning Rate	1e-3

Table 6: Hyperparameters of CoMMA.