How Does Distilled Data Complexity Impact the Quality and Confidence of Non-Autoregressive Machine Translation?

Weijia Xu[*] † **Shuming Ma**‡ **Dongdong Zhang**‡ **Marine Carpuat**†
†Department of Computer Science, University of Maryland
‡Microsoft Research Asia
{weijia, marine}@cs.umd.edu, {shumma, dozhang}@microsoft.com

Abstract

While non-autoregressive (NAR) models are showing great promise for machine translation (MT), their use is limited by their dependence on knowledge distillation from autoregressive models. To address this issue, we seek to understand why distillation is so effective. Prior work suggests that distilled training data is less complex than manual translations. Based on experiments with the Levenshtein Transformer and the Mask-Predict NAR models on the WMT14 German-English task, this paper shows that different types of complexity have different impacts: while reducing lexical diversity and decreasing reordering complexity both help NAR learn better alignment between source and target, and thus improve translation quality, lexical diversity is the main reason why distillation increases model confidence, which affects the calibration of different NAR models differently.

1 Introduction and Background

When training NAR models for neural machine translation (NMT), sequence-level knowledge distillation (Kim and Rush, 2016) is key to match the translation quality of autoregressive (AR) models (Gu et al., 2018; Lee et al., 2018; Ghazvininejad et al., 2019; Gu et al., 2019). Knowledge distillation was first proposed to obtain small *student* models that match the quality of a higher-capacity *teacher* models (Liang et al., 2008; Hinton et al., 2015). Sequence-level knowledge distillation (SLKD) trains the student model $p(\boldsymbol{y} \mid \boldsymbol{x})$ to approximate the teacher distribution $q(\boldsymbol{y} \mid \boldsymbol{x})$ by maximizing the following objective: $\mathcal{L}_{\text{SEQ-KD}} = -\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y} \mid \boldsymbol{x}) \log p(\boldsymbol{y} \mid \boldsymbol{x}) \approx -\sum_{\boldsymbol{y} \in \mathcal{Y}} \mathbb{1}\left[\boldsymbol{y} = \hat{\boldsymbol{y}}\right] \log p(\boldsymbol{y} \mid \boldsymbol{x})$, where $\mathcal{Y}$ represents the space of all possible target sequences, and $\hat{\boldsymbol{y}}$ is the output from running beam search with the teacher model q.

However, we do not yet have a clear picture for how SLKD impacts NAR training. Ren et al. (2020) show that SLKD reduces the degree of dependency between target tokens. Gu et al. (2018) hypothesize that SLKD reduces the number of modes in the output distribution (alternative translations for a source). This hypothesis was supported by experiments that use multiway parallel data to simulate the modes (Zhou et al., 2019). Zhou et al. (2019) also investigate the impact of data complexity on NAR translation quality – they generate distilled data of varying complexity with AR models of different capacity and show that higher-capacity NAR models require more complex distilled data to achieve better translation quality. They further show that generating distilled references with mixture of experts (Shen et al., 2019) improves NAR translation quality. However, training samples can be complex in different ways, and it remains unclear how different types of data complexity alter the internal working of NAR models and their translation quality. We also anticipate that data complexity may impact the uncertainty and calibration of NAR models – an understudied question, unlike for AR models (Ott et al., 2018; Wang et al., 2020).

This paper focuses on two types of data complexity – lexical diversity and degree of word reordering. We expose two state-of-the-art NAR models (Mask-Predict (Ghazvininejad et al., 2019) and Levenshtein Transformer (Gu et al., 2019)) to distilled references of varying complexity on the WMT14 German-English task. Experiments show that decreasing reordering complexity and reducing lexical diversity via distillation both help NAR models learn better alignment between source and target and thus improve translation quality. Further analysis shows that knowledge distillation lowers model uncertainty by reducing lexical diversity, which affects the calibration of Mask-Predict and Levenshtein Transformer models in opposite directions.

[*] Work done during internship at Microsoft Research Asia.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4392–4400
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Generating Diverse Distilled References

We measure **distilled corpus complexity** with:

- **Word Reordering Degree** computed by the average fuzzy reordering score (FRS) (Talbot et al., 2011) over all sentence pairs. FRS is an MT evaluation metric introduced to distinguish significant changes in reordering rules of MT systems on syntactically distant language pairs. A higher FRS indicates that the hypothesis is more monotonically aligned to the source. Zhou et al. (2019) show that distilled data has a higher FRS than the real data which may benefit NAR models.
- **Lexical Diversity** which captures the diversity of target word choices given a source word. We compute the lexical diversity $LD(d)$ of the distilled corpus d by averaging the entropy of target words y conditioned on a source word x (Zhou et al., 2019): $LD(d) = \frac{1}{|\mathcal{V}_x|} \sum_{x \in \mathcal{V}_x} \mathbb{H}\left[y \,|\, x\right]$, where $\mathcal{V}_x$ denotes the source vocabulary.

To isolate the impact of complexity factors, we seek to control the **faithfulness** $F(d)$ of the distilled data d to the real parallel data r. We compute it as the KL-divergence of the alignment distribution between the real data r and the distilled data d (Zhou et al., 2019): $F(d) = \frac{1}{|\mathcal{V}_x|} \sum_{x \in \mathcal{V}_x} D_{\mathrm{KL}}\left[\, p_r(y \,|\, x) || \, p_d(y \,|\, x)\right]$.

Distilled Sample Generation To encourage diversity according to the corpus-level metrics above, we select distilled references for each source from the k-best list of AR hypotheses,[1] using instantiations of the following score:

$$\mathrm{score}(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{y}) = \lambda\, \mathrm{sim}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + (1 - \lambda)\, \mathrm{cxty}(\hat{\boldsymbol{y}}, \boldsymbol{x})$$

where the similarity $\mathrm{sim}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ measures how faithful the hypothesis $\hat{\boldsymbol{y}}$ is to the original reference $\boldsymbol{y}$ and the complexity $\mathrm{cxty}(\hat{\boldsymbol{y}}, \boldsymbol{x})$ captures the relationship between the target sequence $\hat{\boldsymbol{y}}$ and source sequence $\boldsymbol{x}$. The similarity function is the smoothed sentence-level BLEU (Chen and Cherry, 2014) w.r.t the original reference. We use three different complexity functions: 1) **FRS**, 2) **word-alignment score**[2] that measures complexity on a word level, and 3) **NMT score**[3] that measures complexity on a sentence level.

	Real	Distill	Δ
Original	24.2	26.6	+2.4
Reordered	30.0	29.4	−0.6

Table 1: BLEU scores on the original WMT14 En-De and the synthetic reordered version. For each task, we compare LevT models trained on real vs. distilled data.

3 Experimental Settings

Set-Up We use En-De and De-En datasets from WMT14 (Bojar et al., 2014) with the same preprocessing steps as Gu et al. (2019). We evaluate translation quality with case-sensitive tokenized BLEU,[4] using the Moses tokenizer.

Models We use two state-of-the-art NAR models:

- **Mask-Predict (MaskT)** (Ghazvininejad et al., 2019) uses a masked language model (Devlin et al., 2019) to generate the target sequence by iteratively masking out and regenerating the subset of tokens that the model is least confident about.
- **Levenshtein Transformer (LevT)** (Gu et al., 2019) generates the target sequence through iterative insertion and deletion steps.

All AR and NAR models adopt the *base* Transformer architecture (Vaswani et al., 2017). We train all models using a batch size of $64,800$ tokens for maximum $300,000$ steps and select the best checkpoint based on validation perplexity (see Appendix for details). During inference, we set the maximum number of iterations to 10. All word alignments in this paper are generated automatically using *fast-align* (Dyer et al., 2013).[5]

4 Preliminary: SLKD Helps NAR Learn Word Alignment

Our work is motivated by the hypothesis that SLKD helps NAR models learn (implicit) alignment between source and target words. We first test

[1] This is inspired by sequence-level interpolation (Kim and Rush, 2016), but they select hypothesis using BLEU while we use more diverse criteria. We use beam search with $k = 32$.

[2] Sum of the log probabilities of each target word conditioned on its aligned source words given by *fast-align*.

[3] Log probability of the target sentence conditioned on the source given by an AR model.

[4] https://github.com/pytorch/fairseq/blob/master/fairseq/clib/libbleu/libbleu.cpp

[5] This might introduce alignment errors leading to lower absolute FRS scores than with if we had access to gold manual alignments. However, this measurement noise is unlikely to impact our findings because 1) it is likely to be small on distilled data generated by autoregressive NMT models, which should be easier to align than original translations, and 2) distilled data versions are expected to be impacted uniformly.

	Data Property			BLEU ↑	
Data Version	FRS	LexDiv	Faith	MaskT	LevT
Real	0.46	0.36	0.0	28.0	27.6
Distilled	0.55	0.18	7.9	29.6	30.6
selection via BLEU					
+0.5 NMT	0.55	0.17	7.6	29.5	30.6
+0.5 w-align	0.57	0.18	7.6	29.2	30.1 ↓
+0.5 FRS	0.61	0.19	7.6	28.8 ↓	29.6 ↓
selection via BLEU					
+0.2 NMT	0.55	0.17	7.8	29.2	30.4
+0.2 w-align	0.58	0.18	7.9	28.7 ↓	30.0 ↓
+0.2 FRS	0.64	0.19	7.8	28.5 ↓	29.7 ↓

Table 2: Translation quality on WMT14 De-En. In the bottom two groups, models are trained on distilled data with similar faithfulness (*Faith*) but varying degree of reordering (FRS) and lexical diversity (*LexDiv*). ↓ marks significant drops compared to the first row in each group based on the paired bootstrap test at $p < 0.05$ (Clark et al., 2011).

this hypothesis by evaluating the effect of SLKD on two datasets: a) En-De train/dev/test sets from WMT14, and b) a synthetic version of the same task, where word alignment information is embedded by pre-reordering the source words so that they are monotonically aligned with target words (in train/dev/test sets).

While SLKD improves BLEU by +2.4 on the original En-De task, it has no benefit on the synthetic task (Table 1). This supports our hypothesis and is consistent with other findings on real data: Ghazvininejad et al. (2019) and Gu et al. (2019) showed that SLKD improves the quality of NAR models more on syntactically distant language pairs such as German-English than on Romanian-English. Furthermore, Ran et al. (2019) showed that automatically pre-reordering the source words improves the translation quality of NAR models. However, unlike in our experiment, SLKD is still needed in real translation scenarios, as exactly preordering the source is not feasible at test time. Thus, we turn to understanding how distilled data helps NAR models on real translation tasks.

5 Reduced Lexical Diversity in SLKD Improves Translation Quality

We have shown that, similar to the effect of pre-reordering, SLKD benefits NAR training by reducing the difficulty of learning the source-target alignment. However, apart from the word reordering degree, reducing the lexical diversity on the target side can also reduce the difficulty of learning the alignment. In this section, we investigate how the two types of data complexity affect how well NAR models capture the source-target alignment, and therefore translation quality.

	Acc	Conf	ECE ↓
AR Transformer	63.9	72.3	10.34
MaskT w/o SLKD	63.7	74.2	10.49
MaskT w/ SLKD	65.1	86.5	21.41
LevT w/o SLKD	66.8	53.3	20.26
LevT w/ SLKD	65.9	71.3	15.17

Table 3: Average token-level accuracy (*Acc*), confidence (*Conf*), and inference ECE (ECE) of AR and the two NAR models trained with and without SLKD.

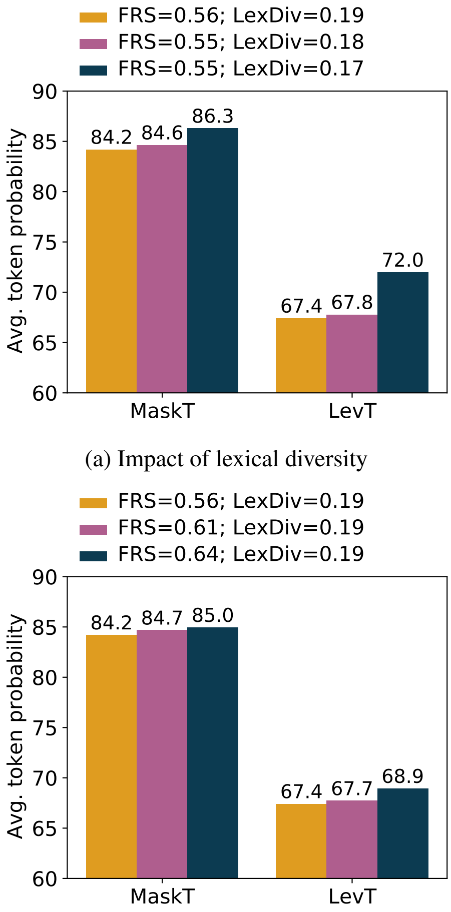
SLKD impacts both complexity types: the first two rows of Table 2 show that SLKD increases FRS by +0.09, reduces lexical diversity by −0.18, and boosts the BLEU of MaskT and LevT by 1.6–3.0 over their counterparts trained on real data.

We then compare NAR models trained on distilled data with varying degree of reordering and lexical diversity while controlling for faithfulness (2nd and 3rd group of rows in Table 2). While the absolute BLEU deltas are small, BLEU decreases significantly as the lexical diversity increases despite reduced degree of reordering. This indicates that increased lexical diversity prevails over the effect of lower degree of reordering in decreasing BLEU scores.

6 SLKD Increases Confidence of Source-Target Attention

To better understand how SLKD helps NAR learn the alignment between source and target, we measure how the *confidence* of the source-target attention changes over decoding iterations. Following Voita et al. (2019), we define the *confidence* of attention heads as the average of the maximum attention weights over source tokens, where the average is taken over target tokens. Higher confidence scores indicate that the model is more certain about which parts of the source sequence to attend to when predicting the target tokens.

As seen in Figure 2, SLKD increases the confidence of source-target attention on both MaskT and LevT. The increase is larger for MaskT than for LevT. For LevT, SLKD increases the attention confidence the most at early decoding iterations. At later iterations, as the model becomes more confident about which source tokens to attend to given the target tokens generated at previous iterations,



(a) Impact of lexical diversity

(b) Impact of word reordering

Figure 1: Average token-level uncertainty of MaskT and LevT trained on distilled data with decreasing degree of lexical diversity (a) and word reordering (b) from yellow to blue.

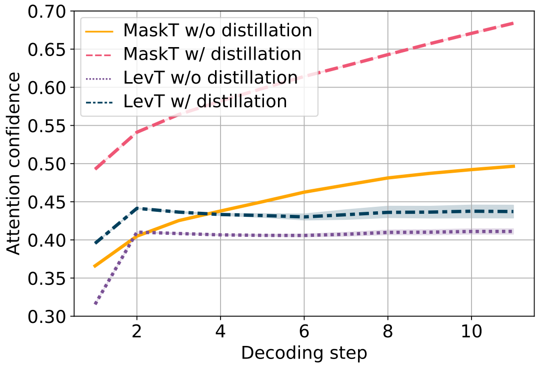
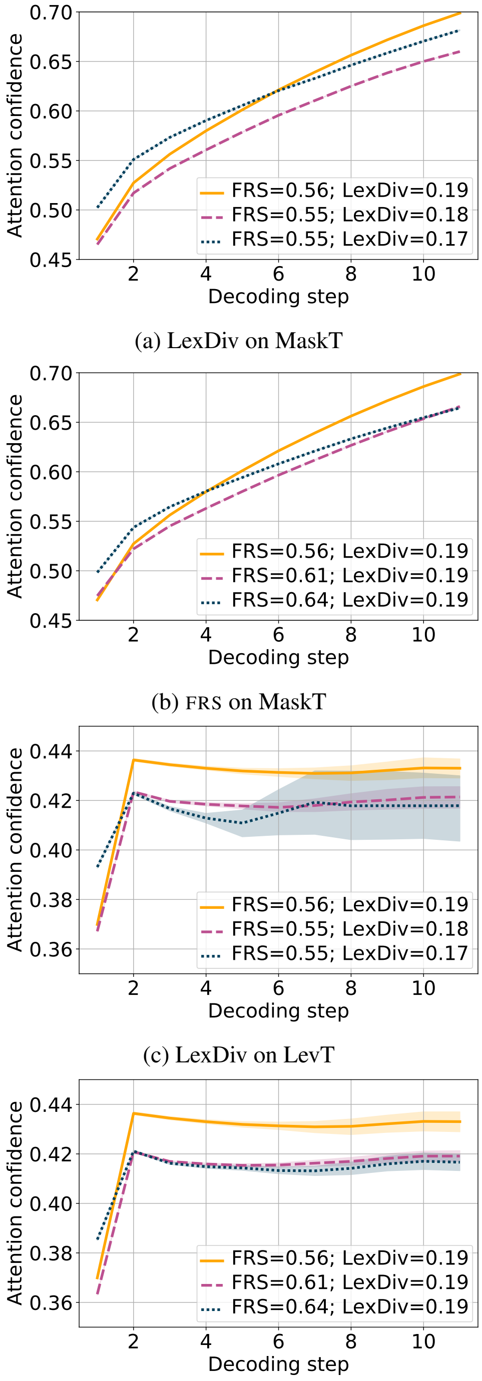


Figure 2: Source-target attention confidence of LevT and MaskT as a function of decoding step.

the impact of SLKD becomes smaller.

Next, we separate the impact of lexical diversity and word reordering (Figure 3). Reducing both types of complexity leads to more concentrated source-target attention at early iterations. By contrast, models trained on more lexically and syntactically diverse data have more distributed source-target attention at iterations, and the attention becomes more concentrated at later iterations as more target tokens have been generated.

Overall, these results suggest that reducing lexical diversity and degree of word reordering both help NAR find the source-target alignment and thus reduce the error rate at the early decoding stage.



(a) LexDiv on MaskT

(b) FRS on MaskT

(c) LexDiv on LevT

(d) FRS on LevT

Figure 3: Source-target attention confidence as a function of decoding step comparing MaskT and LevT trained on distilled data with varying degree of lexical diversity (a, c) and word reordering (b, d).

7 Reduced Lexical Diversity in SLKD Improves Model Confidence

Ott et al. (2018) show that the intrinsic uncertainty of translation – due to the existence of multiple semantically equivalent translations for the same source – is a source of uncertainty in the AR models' output distribution. We hypothesize that these effects might be amplified with NAR models, yet little is known about the confidence and calibration of NAR models. We measure the impact of SLKD on model uncertainty using the average token probability of the models' translation outputs, and the inference Expected Calibration Error (ECE) (Wang et al., 2020) that measures how the model's confidence on a prediction matches to the correctness of the prediction. As shown in Table 3, both MaskT and LevT become more confident when trained with SLKD. However, SLKD causes MaskT to be overconfident and hurts its calibration by $+11\%$ ECE.[6] By contrast, SLKD changes LevT from underconfident to slightly overconfident, improving its calibration by -5% lower ECE.

Next, we isolate the impact of lexical diversity and degree of word reordering on model uncertainty.[7] We measure the average token probability of MaskT and LevT trained on data with varying lexical diversity but close FRS scores (Figure 1a), and vice versa (Figure 1b). Decreasing lexical diversity by -0.02 significantly reduces model uncertainty by 2.1–4.6%, whereas the impact of word reordering degree is small: increasing FRS by $+0.08$ only increases the average uncertainty by 0.8–1.5%. By contrast, SLKD boosts FRS by $+0.09$ over the real data. This suggests that reduced lexical diversity is the main reason why SLKD increases model confidence in lexical choice, which raises concerns since Ding et al. (2021) showed that lexical choice errors are also propagated from AR to NAR models through SLKD.

8 Conclusion

We investigated the effect of knowledge distillation in NAR models trained on distilled data that differs along two types of complexity – lexical diversity and degree of word reordering. Reducing lexical diversity and decreasing word reordering degree both boost the confidence of source-target attention, suggesting that they help NAR models learn the alignment between source and target. Furthermore, distillation increases model confidence by reducing lexical diversity, which improves calibration for LevT but leads to much worse calibration for MaskT. These findings reveal a connection between distillation and existing techniques to improve NAR via pre-reordering (Ran et al., 2019) or integrating external alignment information in the source-target attention (Li et al., 2019).[8]

Our findings are based on experiments on the WMT14 English-German corpus, which is widely used in the literature of NAR translation and has interesting typological properties. While we expect these findings to hold for other tasks that exhibit similar degrees of reordering and lexical diversity, it remains to be seen to what degree they generalize to other language pairs and data settings.

We hope that this work will inspire future research on understanding of the positive and negative impact of knowledge distillation on NAR models, as well as of the more advanced approaches to improving NAR by integrating lexical choice and word reordering knowledge. In addition, our work also calls for future work on improving the calibration of NAR models.

Acknowledgments

We thank the anonymous reviewers, Sweta Agrawal, Naomi Feldman, Zara Harmon, Aquia Richburg, Craig Thorburn, the CLIP lab at UMD for helpful comments, and Eleftheria Briakou for her help on result visualization. This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[6] This might be due to decoding where MaskT repeatedly masks out and re-predicts its least confident predictions.

[7] We only measure their isolated impact on model uncertainty, not ECE, because we could not isolate lexical diversity from degree of word reordering while controlling faithfulness, which impacts ECE through accuracy.

[8] These techniques still rely on knowledge distillation for training, while this paper contributes a systematic study of factors that impact the effectiveness of distillation.

References

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *International Conference on Learning Representations*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

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 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3th International Conference on Learning Representations*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

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. 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 5708–5713, Hong Kong, China. Association for Computational Linguistics.

Percy Liang, Hal Daumé III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 25th international conference on Machine learning*, pages 592–599.

Toan Q. Nguyen and David Chiang. 2018. Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 334–343. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965, Stockholmsmässan, Stockholm Sweden. PMLR.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Computational*, pages 157–163. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2019. Guiding non-autoregressive neural machine translation decoding with reordering information. *CoRR*, abs/1911.02215.

Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. 2020. A study of non-autoregressive model for sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 149–159, Online. Association for Computational Linguistics.

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*, pages 1715–1725. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5719–5728. PMLR.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz Och. 2011. A lightweight evaluation framework for machine translation reordering. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 12–21, Edinburgh, Scotland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2019. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

A Data Preprocessing Details

Following Gu et al. (2019), we preprocess the WMT14 En-De and De-En datasets (Bojar et al., 2014) via normalization, tokenization, true-casing, and joint BPE (Sennrich et al., 2016) with $37K$ operations.[9] The training data contain 3.9M sentence pairs, and the validation and test sets contain $3,000$ and $3,003$ sentence pairs, respectively.

B Model and Training Details

All AR and NAR models adopt the *base* Transformer architecture (Vaswani et al., 2017) with $d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $n_{\text{heads}} = 8$, $n_{\text{layers}} = 6$, and $p_{\text{dropout}} = 0.3$. We tie the source and target embeddings with the output layer weights (Press and Wolf, 2017; Nguyen and Chiang, 2018). We use label smoothing of 0.1. We train the models using Adam (Kingma and Ba, 2015) with initial learning rate of 0.0005 and a batch size of $64,800$ tokens for maximum $300,000$ steps. We select the best checkpoint based on validation perplexity. The total number of parameters is 65M for the AR model, 66M for MaskT, and 91M for LevT. Training takes around 230 hours for each NAR model and 110 hours for each AR model on 4 Tesla P40 GPUs.

C Detailed Experimental Results

Table 4 shows the scores of corpus-level metrics, test BLEU and validation perplexity of MaskT and LevT trained on various distilled versions of WMT14 De-En training data generated through diverse reference generation (Section 2).

D Reference Generation Examples

We show that the k-best list generated by the AR model using beam search is both lexically and syntactically diverse through a random example selected from the training set (Table 5).

[9] Data can be downloaded from `http://dl.fbaipublicfiles.com/nat/original_dataset.zip`

	Data Property			test BLEU		Valid Perplexity	
	FRS	LexDiv	Faith	MaskT	LevT	MaskT	LevT
Real Data	0.46	0.36	0.0	28.0	27.6	35.39	62.49
Distilled Data	0.55	0.18	7.9	29.6	30.6	8.84	11.12
Selection: BLEU	0.54	0.19	7.4	29.4	30.1	9.74	12.57
Selection: BLEU + NMT score ($\lambda = 0.8$)	0.54	0.18	7.4	29.2	30.1	9.45	11.94
Selection: BLEU + NMT score ($\lambda = 0.5$)	0.55	0.17	7.6	29.5	30.6	8.97	11.59
Selection: BLEU + NMT score ($\lambda = 0.2$)	0.55	0.17	7.8	29.2	30.4	8.77	10.94
Selection: BLEU + word-align score ($\lambda = 0.8$)	0.55	0.18	7.4	29.6	30.3	9.63	12.23
Selection: BLEU + word-align score ($\lambda = 0.5$)	0.57	0.18	7.6	29.2	30.1	9.27	11.48
Selection: BLEU + word-align score ($\lambda = 0.2$)	0.58	0.18	7.9	28.7	30.0	8.69	11.24
Selection: BLEU + FRS ($\lambda = 0.8$)	0.56	0.19	7.4	29.1	30.3	9.68	12.10
Selection: BLEU + FRS ($\lambda = 0.5$)	0.61	0.19	7.6	28.8	29.6	9.53	12.25
Selection: BLEU + FRS ($\lambda = 0.2$)	0.64	0.19	7.8	28.5	29.7	8.81	11.71

Table 4: FRS, lexical diversity (*LexDiv*), and faithfulness (*Faith*) scores of various distilled versions of WMT14 De-En training data, test BLEU scores and validation perplexity of MaskT and LevT trained on each data version.

source	Ich hoffe , daß dort in Ihrem Sinne entschieden wird.
original reference	It will , I hope , be examined in a positive light.
translation 1	I hope that it will be decided along your lines.
translation 2	I hope that a decision will be taken along your lines.
translation 3	I hope that the decision will be taken along your lines.
translation 4	I hope that it will be decided in your interest.
translation 5	I hope that there will be a decision along your lines.
translation 6	I hope that decision will be taken along your lines.
translation 7	I hope that the decision will be taken in your interest.
translation 8	I hope that a decision will be taken in your interest.
translation 9	I hope that a decision will be made along your lines.
translation 10	I hope that this will be decided along your lines.
translation 11	I hope that a decision will be taken to that effect.
translation 12	I hope there will be a decision along your lines.
translation 13	I hope that a decision will be taken on your behalf.
translation 14	I hope that a decision will be taken in that regard.
translation 15	I hope that decision will be taken in your interest.
translation 16	I hope that a decision will be taken in that direction.
translation 17	I hope that a decision will be taken in that respect.
translation 18	I hope it will be decided along your lines.
translation 19	I hope that you will take a decision there.
translation 20	I hope that you will take a decision in that regard.
translation 21	I hope that this decision will be taken in your interest.
translation 22	I hope that it will decide along your lines.
translation 23	I hope that it will be decided in your interests.
translation 24	I hope that the decision will be taken in your interests.
translation 25	I hope that the decision will be taken in that direction.
translation 26	I hope that a decision will be taken in your interests.
translation 27	I hope that a decision will be taken to that end.
translation 28	I hope that the decision will be taken in that regard.
translation 29	I hope that a decision will be made in your interest.
translation 30	I hope it will be decided in your interest.
translation 31	I hope that you will take a decision on this.
translation 32	I hope that it will be decided accordingly.

Table 5: An example of the k-best list generated by the AR model using beam search with a beam size of $k = 32$.