Viterbi Decoding of Directed Acyclic Transformer for Non-Autoregressive Machine Translation

Chenze Shao[1,2], **Zhengrui Ma**[1,2], **Yang Feng**[1,2*]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
{shaochenze18z, mazhengrui21b, fengyang}@ict.ac.cn

Abstract

Non-autoregressive models achieve significant decoding speedup in neural machine translation but lack the ability to capture sequential dependency. Directed Acyclic Transformer (DA-Transformer) was recently proposed to model sequential dependency with a directed acyclic graph. Consequently, it has to apply a sequential decision process at inference time, which harms the global translation accuracy. In this paper, we present a Viterbi decoding framework for DA-Transformer, which guarantees to find the joint optimal solution for the translation and decoding path under any length constraint. Experimental results demonstrate that our approach consistently improves the performance of DA-Transformer while maintaining a similar decoding speedup.[1]

1 Introduction

Non-autoregressive translation (Gu et al., 2018) models achieve a significant decoding speedup but suffer from performance degradation, which is mainly attributed to the multi-modality problem. Multi-modality refers to the scenario where the same source sentence may have multiple translations with a strong cross-correlation between target words. However, non-autoregressive models generally hold the conditional independence assumption on target words, which prevents them from capturing the multimodal target distribution.

Recently, Directed Acyclic Transformer (Huang et al., 2022) was proposed to model sequential dependency with a directed acyclic graph consisting of different decoding paths that enable the model to capture multiple translation modalities. Although it has been proven effective, it cannot directly find the most probable translation with the argmax operation. Therefore, DA-Transformer has to apply a sequential decision process at inference time, which harms the global translation accuracy.

In this paper, we propose a Viterbi decoding (Viterbi, 1967) framework for DA-Transformer to improve the decoding accuracy. Using the Markov property of decoding path, we can apply Viterbi decoding to find the most probable path, conditioned on which we can generate the translation with argmax decoding. Then, we further improve this decoding algorithm to perform a simultaneous search for decoding paths and translations, which guarantees to find the joint optimal solution under any length constraint. After Viterbi decoding, we obtain a set of translations with different lengths and rerank them to obtain the final translation. We apply a length penalty term in the reranking process, which prevents the generation of empty translation (Stahlberg and Byrne, 2019) and enables us to control the translation length flexibly.

Experimental results on several machine translation benchmark tasks (WMT14 En↔De, WMT17 Zh↔En) show that our approach consistently improves the performance of DA-Transformer while maintaining a similar decoding speedup.

2 Preliminaries: DA-Transformer

2.1 Model Architecture

DA-Transformer is formed by a Transformer encoder and a directed acyclic decoder. The encoder and layers of the decoder are the same as vanilla Transformer (Vaswani et al., 2017). On top of the decoder, the hidden states are organized as a directed acyclic graph, whose edges represent transition probabilities between hidden states.

Given a source sentence $X = \{x_1, \cdots, x_N\}$ and a target sentence $Y = \{y_1, \cdots, y_M\}$, the decoder length L is set to $\lambda \cdot N$, where λ is a hyperparameter. The translation probability from X to

[*]Corresponding author: Yang Feng

[1]We implement our method on https://github.com/thu-coai/DA-Transformer.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4390–4397
December 7-11, 2022

Y is formulated as:

$$P_\theta(Y|X) = \sum_{A\in\Gamma} P_\theta(A|X) P_\theta(Y|X, A), \quad (1)$$

where $A = \{a_1, \cdots, a_M\}$ is a translation path for the target sentence Y and a_i represents the position of word y_i in the decoder. Γ contains all possible translation paths with $1 = a_1 < \cdots < a_M = L$.

The probability of translation path A is formulated based on the Markov hypothesis:

$$P_\theta(A|X) = \prod_{i=1}^{M-1} P_\theta(a_{i+1}|a_i, X) = \prod_{i=1}^{M-1} E_{a_i,a_{i+1}}, \quad (2)$$

where $E \in \mathbb{R}^{L\times L}$ is the transition matrix obtained by self-attention, and $E_{a_i,a_{i+1}}$ represents the transition probability from position a_i to position a_{i+1}. E is masked by a lower triangular matrix to ensure that the translation path is acyclic.

Conditioned on X and the translation path A, the translation probability of Y is formulated as:

$$P_\theta(Y|A, X) = \prod_{i=1}^{M} P_\theta(y_i|a_i, X), \quad (3)$$

where $P_\theta(y_i|a_i, X)$ represents the translation probability of word y_i on the position a_i of decoder.

2.2 Training and Inference

The training objective of DA-Transformer is to maximize the log-likelihood $\log P_\theta(Y|X)$, which requires marginalizing all paths A. Using the Markov property of translation path, DA-Transformer employs dynamic programming to calculate the translation probability. Besides, it applies glancing training (Qian et al., 2021) with a hyper-parameter τ to promote the learning.

During inference, the objective is to find the most probable translation $\mathrm{argmax}_Y\, P_\theta(Y|X)$. However, there is no known tractable decoding algorithm for this problem. Huang et al. (2022) proposed three approximate decoding strategies to find high-probability translations. The intuitive strategy is greedy decoding, which sequentially takes the most probable transition as the decoding path and generates a translation according to the conditional probabilities. Lookahead decoding improves greedy decoding by taking the most probable combination of transition and prediction as follows:

$$y_i^*, a_i^* = \underset{y_i,a_i}{\mathrm{argmax}}\, P_\theta(y_i|a_i, X) P_\theta(a_i|a_{i-1}, X). \quad (4)$$

Beam search decoding is a more accurate method that merges the paths of the same prefix, which approximates the real translation probability and better represents the model's preference. Beam search can be optionally combined with an n-gram language model to improve the performance further. However, the speed of beam search is much lower than greedy and lookahead decoding.

3 Methodology

This section presents a Viterbi decoding framework for DA-Transformer to improve decoding accuracy. We first develop a basic algorithm to find the optimal decoding path and then improve it to find the joint optimal solution of the translations and decoding paths. Finally, we introduce the technique to rerank the Viterbi decoding outputs.

3.1 Optimal Decoding Path

Recall that the greedy decoding strategy sequentially takes the most probable transition as the decoding path, which may not be optimal since the greedy strategy does not consider long-term profits. In response to this problem, we propose a Viterbi decoding framework for DA-Transformer that guarantees to find the optimal decoding path $\mathrm{argmax}_A\, P_\theta(A|X)$ under any length constraint.

Specifically, we consider decoding paths of length i that end in position $a_i = t$, and use $\alpha(i,t)$ to represent the maximum probability of these paths. By definition, we set the initial state $\alpha(1,1)=1$ and $\alpha(1,t>1)=0$. The Markov property of decoding paths enables us to sequentially calculate $\alpha(i,\cdot)$ from its previous step $\alpha(i-1,\cdot)$:

$$\begin{aligned} \alpha(i,t) &= \max_{t'} \alpha(i-1,t') \cdot E_{t,t'}, \\ \psi(i,t) &= \underset{t'}{\mathrm{argmax}}\, \alpha(i-1,t') \cdot E_{t,t'}, \end{aligned} \quad (5)$$

where E is the transition matrix defined in Equation 2 and $\psi(i,t)$ is the backtracking index pointing to the previous position. After L iterations, we obtain the score for every possible length, and then we can find the optimal length with the argmax function:

$$M = \underset{i}{\mathrm{argmax}}\, \alpha(i, L). \quad (6)$$

After determining the length M, we can trace the best decoding path along the backtracking index starting from $a_M = L$:

$$a_i = \psi(i+1, a_{i+1}). \quad (7)$$

Finally, conditioning on the optimal path A, we can generate the translation with argmax decoding:

$$y_i = \underset{y_i}{\operatorname{argmax}} P_\theta(y_i|a_i, X). \tag{8}$$

3.2 Joint Optimal Solution

The decoding algorithm described above can be summarized as the following process:

$$\begin{aligned} A^* &= \underset{A}{\operatorname{argmax}} P_\theta(A|X), \\ Y^* &= \underset{Y}{\operatorname{argmax}} P_\theta(Y|X, A^*). \end{aligned} \tag{9}$$

Even though the algorithm now finds the optimal decoding path, the translation on this path may have low confidence, resulting in a low joint probability $P_\theta(A, Y|X)$. We further improve the decoding algorithm to search for both decoding paths and translations, which guarantees to find the joint optimal solution:

$$A^*, Y^* = \underset{A,Y}{\operatorname{argmax}} P_\theta(A, Y|X). \tag{10}$$

Notice that when the path A is given, we can easily find the most probable translation Y with argmax decoding. Let Y^A denotes the argmax decoding result under path A, where $y_i^{a_i} = \operatorname{argmax}_{y_i} P_\theta(y_i|a_i, X)$ is the i-th word of Y^A. Then we can simplify our objective with Y^A:

$$\begin{aligned} &\max_{A,Y} P_\theta(A,Y|X) = \max_{A,Y} P_\theta(A|X)P_\theta(Y|X,A) \\ &= \max_A (P_\theta(A|X) \max_Y P_\theta(Y|X,A)) \\ &= \max_A P_\theta(A|X)P_\theta(Y^A|X, A) \\ &= \max_A P_\theta(y_1^{a_1}|a_1,X) \prod_{i=1}^{M-1} E_{a_i,a_{i+1}} P_\theta(y_{i+1}^{a_{i+1}}|a_{i+1},X) \\ &= \max_A P_\theta(y_1^A|a_1, X) \prod_{i=1}^{M-1} {E'}_{a_i,a_{i+1}}, \end{aligned} \tag{11}$$

where we introduce a new transition matrix E' with $E'_{a_i,a_{i+1}} = E_{a_i,a_{i+1}} P_\theta(y_{i+1}^{a_{i+1}}|a_{i+1}, X)$. Compared to $\max_A P_\theta(A|X)$, the major difference is the transition matrix E', which considers both the transition probability and the prediction probability. Therefore, we can still apply the Viterbi decoding framework to find the optimal joint solution.

We use 'Viterbi' to represent the Viterbi decoding algorithm proposed in section 3.1, and use 'Joint-Viterbi' to represent the improved algorithm in this section that finds the joint optimal solution. It is worth noting that Viterbi and Joint-Viterbi can be regarded as improvements to greedy decoding and lookahead decoding, respectively. Both greedy decoding and lookahead decoding consider the one-step probability and find the next token with $\operatorname{argmax}_{a_i} P_\theta(a_i|X, a_{i-1})$ and $\operatorname{argmax}_{y_i,a_i} P_\theta(y_i|a_i, X)P_\theta(a_i|a_{i-1}, X)$, respectively. In comparison, Viterbi and Joint-Viterbi consider the whole decoding path and guarantee to find the global optimal solution $\operatorname{argmax}_A P_\theta(A|X)$ and $\operatorname{argmax}_{A,Y} P_\theta(A, Y|X)$, respectively.

3.3 Reranking with Length Penalty

After Viterbi decoding, we have a set of translations of different lengths that can be ranked to obtain the most probable one. However, argmax decoding is biased toward short translations and may even degenerate to an empty translation, as also observed in Stahlberg and Byrne (2019).

To solve this problem, we introduce the hyper-parameter β for length normalization in Wu et al. (2016) and modify Equation 6 to divide by the length penalty term:

$$M = \underset{i}{\operatorname{argmax}} \frac{\alpha(i, L)}{i^\beta}. \tag{12}$$

By changing the length penalty β to different values, we now have the flexibility to control the translation length with little additional overhead, which is another appealing feature of our approach.

4 Experiments

4.1 Settings

We conduct experiments on WMT14 English↔German (En↔De, 4.5M pairs) and WMT17 Chinese↔English (Zh↔En, 20M pairs). These datasets are all encoded into subword units (Sennrich et al., 2016). We use the same preprocessed data and train/dev/test splits as Kasai et al. (2020). The translation quality is evaluated with sacreBLEU (Post, 2018) for WMT17 En-Zh and tokenized BLEU (Papineni et al., 2002) for other benchmarks. We use GeForce RTX 3090 to train models and measure translation latency. Our models are implemented based on the open-source toolkit of fairseq (Ott et al., 2019).

We strictly follow the hyper-parameter settings of Huang et al. (2022) to reimplement DA-Transformer. We adopt Transformer-base (Vaswani et al., 2017) as the model architecture. We set dropout to 0.1, weight decay to 0.01, and label smoothing to 0.1 for regularization. We use $\lambda = 8$

Models	Iter	WMT14 En-De	WMT14 De-En	WMT17 En-Zh	WMT17 Zh-En	Average Gap	Speedup
Transformer	M	27.67	31.84	35.05	24.26	0	1.0×
DA-Transformer + Greedy	1	26.06	30.69	33.29	22.32	1.62	14.2×
DA-Transformer + Viterbi	1	26.43†	30.84	33.25	22.58†	1.43	13.3×
DA-Transformer + Lookahead	1	26.55	30.81	33.54	22.68	1.31	14.0×
DA-Transformer + Joint-Viterbi	1	26.89†	31.10†	33.65	23.24†	0.98	13.2×

Table 1: Results on WMT14 En↔De and WMT17 Zh↔En. M is the length of the target sentence. 'Iter' means the number of decoding iterations. The speedup is evaluated on WMT14 En-De test set with a batch size of 1. † means significantly better than the baseline model ($p < 0.05$). We use the statistical significance test with paired bootstrap resampling (Koehn, 2004).

for the graph size and linearly anneal τ from 0.5 to 0.1 for the glancing training. For fair comparisons, we tune the length penalty in $[0.95, 1.05]$ to obtain a similar translation length as lookahead. We train all models for 300K steps, where each batch contains approximately 64K source tokens. All models are optimized by Adam (Kingma and Ba, 2014) with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. The learning rate warms up to $5 \cdot 10^{-4}$ and then begins to anneal it after 10K steps with the inverse square-root schedule. We calculate the validation BLEU scores every epoch and obtain the final model by taking an average of the best five checkpoints.

4.2 Main Results

As shown in Table 1, both Viterbi and Joint-Viterbi improve over their corresponding baseline. Joint-Viterbi achieves the best performance, which outperforms the previous lookahead strategy by 0.33 BLEU. Besides, it is worth noting that the Viterbi decoding process is highly parallelizable, which does not bring much overhead in the decoding and only reduces the speedup by less than $1\times$.

4.3 Results with Knowledge Distillation

In this section, we evaluate the performance of our method with sequence-level knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016), where the target side of the training set is replaced by the output of an autoregressive teacher model. Experimental results in Table 2 show that the differences between decoding strategies are relatively small.

Intuitively, we attribute this phenomenon to the improvement of model confidence. As knowledge distillation reduces the multi-modality of the dataset (Zhou et al., 2020; Sun and Yang, 2020), the model may become more confident in predicting target sentences, which makes the greedy strategy more likely to reach the optima. To verify this,

Method	Greedy	Lookahead	Viterbi	Joint-Viterbi
BLEU	26.81	26.91	26.88	27.03

Table 2: Results with knowledge distillation on WMT14 En-De test set.

Metric	T-Entropy	P-Entropy	Percentage
w/o kd	1.088	1.892	59.6%
w/ kd	0.998	0.601	70.1%

Table 3: Statistics of DA-Transformer on WMT14 En-De test set. 'kd' means knowledge distillation. 'T-' means transition and 'P-' means prediction.

we measure the average entropy of transition and prediction probabilities and evaluate the percentage of lookahead outputs that match the optima $\mathrm{argmax}_{A,Y} P(A, Y|X)$ under their length. As Table 3 shows, DA-Transformer with distillation has smaller entropies and a larger percentage of optimal translations, which confirms our intuition.

4.4 Probability Analysis

Recall that the decoding objective is to find the most probable translation $\mathrm{argmax}_Y P(Y|X)$, while our approach finds the joint solution $\mathrm{argmax}_{A,Y} P(A, Y|X)$. Although there is a gap between them, we argue that optimizing the joint probability helps us achieve higher translation probability. To prove it, we collect the outputs of lookahead decoding and Joint-Viterbi on WMT14 En-De test set and compute their probabilities $P(Y|X)$ by dynamic programming. We then calculate the average log probability of each decoding strategy, and also evaluate the percentage of translations that one strategy obtains a larger probability than another. As Table 4 shows, Joint-Viterbi outperforms lookahead decoding by a large margin, indicating that we can obtain a higher average translation probability

Method	Lookahead	Joint-Viterbi
Log-prob	−4.39	−4.14
Percentage	24.4%	41.6%

Table 4: Probability analysis of on Lookahead and Joint-Viterbi decoding on WMT14 En-De test set.

by optimizing the joint probability.

4.5 Effect of Length Penalty

Viterbi decoding is capable of flexibly controlling the output length with the length penalty β. To show the effect of the length penalty, we change the value of β in Joint-Viterbi to decode the WMT17 Zh-En test set and report the corresponding BLEU scores and average output lengths in Figure 1. It shows that the length penalty can almost linearly control the output length, which can help us obtain satisfactory translations. Generally, Viterbi decoding can obtain better performance when the output length is closer to the reference length. If there is no length penalty, only finding outputs with the maximum joint probability will break the translation quality with extremely small output lengths.

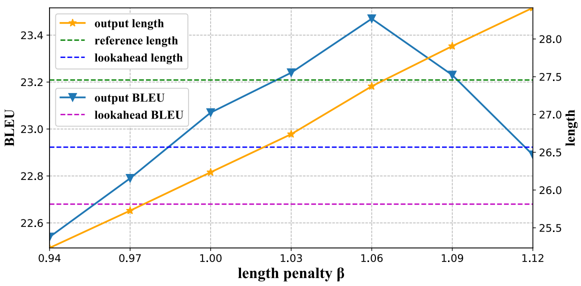

Figure 1: The effect of length penalty β measured on WMT17 Zh-En test set.

5 Related Works

Most non-autoregressive models can directly find the most probable output with argmax decoding, which is the fastest decoding algorithm. However, models of this type usually suffer from the multi-modality problem (Gu et al., 2018), leading to severe performance degradation. A relatively more accurate method is noisy parallel decoding, which requires generating multiple translation candidates and greatly increases the amount of computation.

Many efforts have been made to address the multi-modality problem, including latent models (Kaiser et al., 2018; Ma et al., 2019; Shu et al., 2020; Bao et al., 2021, 2022), alignment-based models (Gu et al., 2018; Ran et al., 2021; Song et al., 2021), and better training objectives (Shao et al., 2019, 2020; Shan et al., 2021; Ghazvininejad et al., 2020; Du et al., 2021; Shao et al., 2021). However, these techniques are still not powerful enough, which heavily rely on knowledge distillation (Kim and Rush, 2016).

Some researchers seek iterative decoding approaches to improve translation quality. Work in this area includes semi-autoregressive decoding (Wang et al., 2018), iterative refinement (Lee et al., 2018), mask-predict decoding (Ghazvininejad et al., 2019), Levenshtein Transformer (Gu et al., 2019), multi-thread decoding (Ran et al., 2020), Imputer (Saharia et al., 2020), and rewriting (Geng et al., 2021). Although their translations are of better quality, they are criticized for being slow at inference time (Kasai et al., 2021).

Recently, latent alignment models like CTC (Libovický and Helcl, 2018; Saharia et al., 2020) and DA-Transformer (Huang et al., 2022) achieved impressive performance and received a lot of attention. Beam search is an useful decoding strategy for latent alignment models (Kasner et al., 2020; Gu and Kong, 2020; Zheng et al., 2021; Shao et al., 2022; Huang et al., 2022; Shao and Feng, 2022). It brings considerable improvements but also reduces the decoding speed.

Viterbi decoding has also been used in non-autoregressive models. In CRF-based NAT models, Viterbi decoding is applied to find the most probable output (Sun et al., 2019; Sun and Yang, 2020).

6 Conclusion

The current decoding strategies of DA-Transformer need to apply a sequential decision process, which harms the global translation accuracy. In this paper, we propose a Viterbi decoding framework for DA-Transformer to find the joint optimal solution of the translation and decoding path and further demonstrate its effectiveness on multiple benchmarks.

7 Acknowledgement

We thank the anonymous reviewers for their insightful comments. We thank Fei Huang for helping us open source code.

Limitations

The major limitation of our method is that it cannot find the most probable translation

$\text{argmax}_Y P(Y|X)$ but alternatively finds the joint optimal solution $\text{argmax}_{A,Y} P(A, Y|X)$. However, as we show in section 4.4, outputs with higher joint probability usually also have higher translation probability, suggesting that optimizing the joint probability is helpful.

Another limitation is that the improvements of our method are smaller in the knowledge distillation setting. However, the main advantage of DA-Transformer is that it does not heavily rely on knowledge distillation and achieves superior performance on raw data, which makes the impact of this limitation small.

References

Yu Bao, Shujian Huang, Tong Xiao, Dongqi Wang, Xinyu Dai, and Jiajun Chen. 2021. Non-autoregressive translation by learning target categorical codes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5749–5759, Online. Association for Computational Linguistics.

Yu Bao, Hao Zhou, Shujian Huang, Dongqi Wang, Lihua Qian, Xinyu Dai, Jiajun Chen, and Lei Li. 2022. latent-GLAT: Glancing at latent variables for parallel text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8398–8409, Dublin, Ireland. Association for Computational Linguistics.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *ICML*.

Xinwei Geng, Xiaocheng Feng, and Bing Qin. 2021. Learning to rewrite for non-autoregressive neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3297–3308, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *ICML*.

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.

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

Jiatao Gu and Xiang Kong. 2020. Fully non-autoregressive neural machine translation: Tricks of the trade.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. 2022. Directed acyclic transformer for non-autoregressive machine translation. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2390–2399. PMLR.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. In *ICML*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zdeněk Kasner, Jindřich Libovický, and Jindřich Helcl. 2020. Improving fluency of non-autoregressive machine translation.

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. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. 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 4282–4292, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2020. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3059–3069, Online. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2021. Guiding non-autoregressive neural machine translation decoding with reordering information. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 13727–13735. AAAI Press.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, 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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yong Shan, Yang Feng, and Chenze Shao. 2021. Modeling coverage for non-autoregressive neural machine translation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Chenze Shao and Yang Feng. 2022. Non-monotonic latent alignments for ctc-based non-autoregressive machine translation. In *Proceedings of NeurIPS 2022*.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. 2019. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024, Florence, Italy. Association for Computational Linguistics.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, and Jie Zhou. 2021. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. *Computational Linguistics*, 47(4):891–925.

Chenze Shao, Xuanfu Wu, and Yang Feng. 2022. One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3779–3791, Seattle, United States. Association for Computational Linguistics.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 198–205. AAAI Press.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *AAAI*.

Jongyoon Song, Sungwon Kim, and Sungroh Yoon. 2021. AligNART: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1–14, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems 32*, pages 3016–3026.

Zhiqing Sun and Yiming Yang. 2020. An EM approach to non-autoregressive conditional sequence generation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9249–9258. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488, Brussels, Belgium. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Jiajun Chen, Jingjing Xu, and Lei Li. 2021. Duplex sequence-to-sequence learning for reversible machine translation.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.