AligNART: Non-autoregressive Neural Machine Translation by Jointly Learning to Estimate Alignment and Translate

Jongyoon Song[1,2*]   **Sungwon Kim**[1]   **Sungroh Yoon**[1,3†]

[1]Data Science and AI Laboratory, Seoul National University, South Korea
[2]Kakao Enterprise, South Korea
[3]Interdisciplinary Program in Artificial Intelligence, Seoul National University, South Korea
{coms1580, ksw0306, sryoon}@snu.ac.kr

Abstract

Non-autoregressive neural machine translation (NART) models suffer from the *multi-modality problem* which causes translation inconsistency such as token repetition. Most recent approaches have attempted to solve this problem by implicitly modeling dependencies between outputs. In this paper, we introduce AligNART, which leverages full alignment information to explicitly reduce the modality of the target distribution. AligNART divides the machine translation task into (i) alignment estimation and (ii) translation with aligned decoder inputs, guiding the decoder to focus on simplified one-to-one translation. To alleviate the alignment estimation problem, we further propose a novel *alignment decomposition* method. Our experiments show that AligNART outperforms previous non-iterative NART models that focus on explicit modality reduction on WMT14 En↔De and WMT16 Ro→En. Furthermore, AligNART achieves BLEU scores comparable to those of the state-of-the-art connectionist temporal classification based models on WMT14 En↔De. We also observe that AligNART effectively addresses the token repetition problem even without sequence-level knowledge distillation.

1 Introduction

In the neural machine translation (NMT) domain, non-autoregressive NMT (NART) models (Gu et al., 2018) have been proposed to alleviate the low translation speeds of autoregressive NMT (ART) models. However, these models suffer from degenerated translation quality (Gu et al., 2018; Sun et al., 2019). To improve the translation quality of NART, several studies on NART iteratively refine decoded outputs with minimal iterations (Ghazvininejad et al., 2019; Kasai et al., 2020a; Lee et al., 2020; Guo et al., 2020; Saharia et al., 2020); other recent

works target to improve NART without iteration (Qian et al., 2021; Gu and Kong, 2021).

One of the significant limitations of non-iterative NART models is the *multi-modality problem*. This problem originates from the fact that the models should maximize the probabilities of multiple targets without considering conditional dependencies between target tokens. For example, in English-to-German translation, a source sentence "Thank you very much." can be translated to "Danke schön." or "Vielen Dank.". Under the conditional independence assumption, the non-iterative NART models are likely to generate improper translations such as "Danke Dank." or "Vielen schön." (Gu et al., 2018). For the same reason, other inconsistency problems such as token repetition or omission occur frequently in non-iterative NART (Gu and Kong, 2021).

There are two main methods for non-iterative NART to address the *multi-modality problem*. Some works focus on an implicit modeling of the dependencies between the target tokens (Gu and Kong, 2021). For example, Ghazvininejad et al. (2020), Saharia et al. (2020), and Gu and Kong (2021) modify the objective function based on dynamic programming, whereas Qian et al. (2021) provide target tokens to the decoder during training.

On the other hand, other works focus on an explicit reduction of the modality of the target distribution by utilizing external source or target sentence information rather than modifying the objective function. For example, Akoury et al. (2019) and Liu et al. (2021) use syntactic or semantic information; Gu et al. (2018), Zhou et al. (2020b), and Ran et al. (2021) use the alignment information between source and target tokens. However, previous explicit modality reduction methods show suboptimal performance.

Zhou et al. (2020b) and Ran et al. (2021) extract fertility (Brown et al., 1993) and ordering

information in word alignments, which enables the modeling of several types of mappings except for many-to-one and many-to-many cases. We hypothesize that leveraging entire mappings significantly reduces the modality and is the key to performance improvement.

In this work, we propose AligNART, a non-iterative NART model that mitigates the *multi-modality problem* by utilizing complete information in word alignments. AligNART divides the machine translation task into (i) alignment estimation and (ii) non-autoregressive translation under the given alignments. Modeling all the type of mapping guides (ii) more close to one-to-one translation. In AligNART, a module called Aligner is simply augmented to NAT (Gu et al., 2018) which estimates alignments to generate aligned decoder inputs.

However, it is challenging to estimate the complex alignment information using only source sentence during inference. Specifically, Aligner should simultaneously predict the number of target tokens corresponding to each source token and their mapping. To overcome this problem, we further propose *alignment decomposition* which factorizes the alignment process into three sub-processes: *duplication*, *permutation*, and *grouping*. Each sub-process corresponds to much feasible sub-problems: one-to-many mapping, ordering, and many-to-one mapping, respectively.

Our experimental results show that AligNART outperforms previous non-iterative NART models of explicit modality reduction on WMT14 En↔De and WMT16 Ro→En. AligNART achieves performance comparable to that of the recent state-of-the-art non-iterative NART model on WMT14 En↔De. We observe that the modality reduction in AligNART addresses the token repetition issue even without sequence-level knowledge distillation (Kim and Rush, 2016). We also conduct quantitative and qualitative analyses on the effectiveness of *alignment decomposition*.

2 Background

Given a source sentence $x = \{x_1, x_2, ..., x_M\}$ and its translation $y = \{y_1, y_2, ..., y_N\}$, ART models with encoder-decoder architecture are trained with chained target distributions and infer the target sentence autoregressively:

$$p(y|x) = \prod_{n=1}^{N} p(y_n|y_{<n}, x). \qquad (1)$$

At each decoding position n, the decoder of the model is conditioned with previous target tokens $y_{<n} = \{y_1, ..., y_{n-1}\}$, which is the key factor of performance in ART models. Previous target tokens reduce the target distribution modality and provide information about the target sentence. However, the autoregressive decoding scheme enforces the decoder to iterate N times to complete the translation and increases the translation time linearly with respect to the length of the target sentence.

Non-iterative NART models (Gu et al., 2018; Sun et al., 2019; Sun and Yang, 2020) assume conditional independence between the target tokens to improve the translation speed:

$$p(y|x) = p(N|x) \cdot \prod_{n=1}^{N} p(y_n|x), \qquad (2)$$

where N is the predicted target length to parallelize the decoding process. Non-iterative NART models provide only the length information of the target sentence to the decoder, which is insufficient to address the *multi-modality problem*.

3 AligNART

3.1 Model Overview

Given the word alignments between the source and target sentences $A \in \{0, 1\}^{N \times M}$, we factorize the task into (i) alignment estimation and (ii) translation with aligned decoder inputs as follows:

$$p(y|x) = p(A|x) \cdot \prod_{n=1}^{N} p(y_n|x, A), \qquad (3)$$

where M and N are the lengths of the source and target sentences, respectively. Although we can also modify the negative log-likelihood loss to model dependencies between outputs such as connectionist temporal classification (CTC) loss (Graves et al., 2006), we focus on the effect of the introduction of alignment as additional information. AligNART is based on the encoder-decoder architecture, with an alignment estimation module called Aligner as depicted in Figure 1a. The encoder maps the embedding of the source tokens into hidden representations $h = \{h_1, h_2, ..., h_M\}$. Aligner constructs the aligned decoder inputs $d = \{d_1, d_2, ..., d_N\}$ as follows:

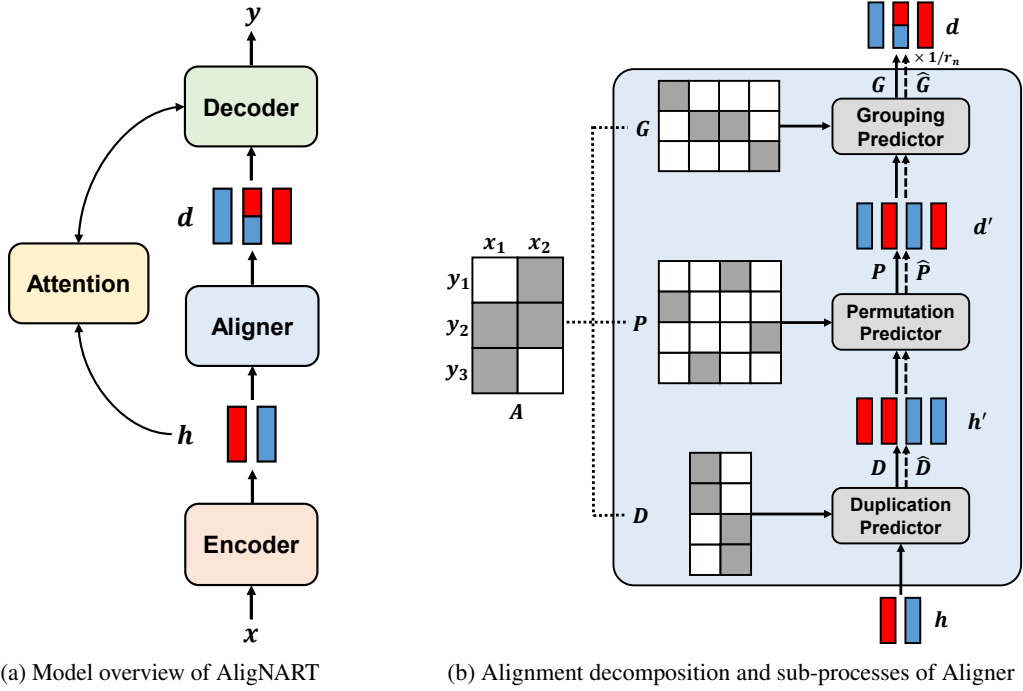$$d_n = \frac{1}{r_n} \sum_{m=1}^{M} A_{n,m} \cdot h_m. \qquad (4)$$

(a) Model overview of AligNART

(b) Alignment decomposition and sub-processes of Aligner

Figure 1: (a) Given the encoder outputs h, Aligner returns aligned encoder outputs d. The decoder then translates the aligned inputs to y. (b) The dotted lines indicate the *alignment decomposition*. During training, predictors are trained with the decomposed matrices D, P, and G, and align inputs using the ground truth as indicated by the solid lines. During inference, predictors align inputs using the estimated matrices as indicated by the dashed lines.

where r_n is the number of non-zero elements in the n-th row of A. Given the aligned decoder inputs, the decoder is guided to focus on a one-to-one translation from d_n to y_n. One-to-one mapping significantly reduces the modality of the target distribution.

The key component of AligNART, Aligner, models a conditional distribution of alignments A given the source sentence x during training, and aligns encoder outputs using the estimated alignments during inference, as depicted in Figure 1b. The ground truth of the alignments is extracted using an external word alignment tool. However, alignment estimation given only the source sentence is challenging since the alignment consists of two components related with target tokens:

- The number of target tokens that correspond to each encoder output h_m.

- The positions of the target tokens to which h_m corresponds.

The Aligner decomposes the alignment for effective estimation, which is described in Section 3.2.

3.2 Aligner

To alleviate the alignment estimation problem, we start by factorizing the alignment process as shown in Figure 1b. First, we copy each encoder output h_m by the number of target tokens mapped to h_m, which is denoted as $c_m = \sum_n A_{n,m}$. Given the duplicated encoder outputs h', we have to predict the positions of target tokens to which each element in h' is mapped.

We further decompose the remaining prediction process into *permutation* and *grouping*, since non-iterative NART models have no information about the target length N during inference. In the permutation process, h' is re-ordered into d' such that elements corresponding to the same target token are placed adjacent to each other. In the grouping process, each element in d' is clustered into N groups by predicting *whether each element is mapped to the same target token as the previous element*. $r_n = \sum_m A_{n,m}$ denotes the number of elements in the n-th group which is equivalent to r_n in Equation 4. Finally, we can derive the decoder inputs d in Equation 4 by averaging the elements in each group in d'. In summary, we decompose the alignment estimation task into three sequential sub-tasks: *duplication*, *permutation*, and *grouping*.

3.2.1 Alignment Decomposition

As shown in Figure 1b, we factorize the alignment matrix A into duplication, permutation, and

3

grouping matrices that correspond to each process. $h' = \{h_{1,1}, ..., h_{1,c_1}, ..., h_{M,1}, ..., d_{M,c_M}\}$ denotes the duplicated encoder outputs where $h_{i,j}$ is the j-th copied element of h_i. Similarly, $d' = \{d_{1,1}, ..., d_{1,r_1}, ..., d_{N,1}, ..., d_{N,r_N}\}$ denotes the permuted encoder outputs where $d_{i,j}$ is the j-th element in the i-th group. The number of non-zero elements in the alignment matrix is defined as $L = \sum_m c_m = \sum_n r_n$.

Duplication Matrix Aligner copies h_m by c_m to construct the duplicated encoder outputs h' with a duplication matrix $D \in \{0,1\}^{L \times M}$. Let $C_m = \sum_{i=1}^{m} c_i$ and $C_0 = 0$. Then, we can define D using c_m as follows:

$$D_{l,m} = \begin{cases} 1 & \text{if } C_{m-1} < l \le C_m \\ 0 & \text{else.} \end{cases} \quad (5)$$

We index h' by the following rule:

- For any $h_{m,i}$ and $h_{m,j}$ ($i < j$), which are matched to d_{x_i,y_i} and d_{x_j,y_j}, respectively, $x_i \le x_j$ and $y_i \le y_j$.

The duplication matrix D contains similar information to fertility (Gu et al., 2018).

Permutation Matrix Aligner re-orders h' to construct d' with a permutation matrix $P \in \{0,1\}^{L \times L}$. Since all the indexed elements in h' and d' are distinct, the permutation matrix P is uniquely defined.

Grouping Matrix Aligner finally aggregates d' to construct d, the aligned decoder inputs, with a grouping matrix $G \in \{0,1\}^{N \times L}$. Let $R_n = \sum_{i=1}^{n} r_i$ and $R_0 = 0$. Then, G can be defined using r_n as follows:

$$G_{n,l} = \begin{cases} 1 & \text{if } R_{n-1} < l \le R_n \\ 0 & \text{else.} \end{cases} \quad (6)$$

We index d' by the following rule:

- For any $d_{n,i}$ and $d_{n,j}$ ($i < j$), which are matched to h_{x_i,y_i} and h_{x_j,y_j}, respectively, $x_i \le x_j$ and $y_i \le y_j$.

We can derive the aligned decoder inputs by separately estimating the decomposed matrices D, P, and G, which approximately correspond to one-to-many mapping, ordering, and many-to-one mapping, respectively. The decomposed matrices have an easily predictable form while recovering the complete alignment matrix.

3.2.2 Training

Aligner consists of three prediction sub-modules: duplication, permutation, and grouping predictors. Each of them estimates the decomposed alignment matrix as follows:

$$p(A|x) = p(G|x, P, D) \cdot p(P|x, D) \cdot p(D|x). \quad (7)$$

The duplication predictor learns to classify the number of copies of h_m. The duplication loss is defined as follows:

$$\mathcal{L}_D = -\frac{1}{M} \sum_{m=1}^{M} \log p_m(c_m), \quad (8)$$

where p_m is the predicted probability distribution of the duplication at the position m. To discriminate copied elements in h', we add copy position embedding to $\{h_{m,1}, ..., h_{m,c_m}\}$ for the next two predictors.

The permutation predictor takes the duplicated encoder outputs h' as inputs. We simplify the permutation prediction problem into a classification of the re-ordered position. For the permutation loss, we minimize the KL-divergence between the prediction P^{pred} and the ground truth P^{GT}.

$$\mathcal{L}_P = -\frac{1}{L} \sum_i \sum_j P_{i,j}^{GT} \log P_{i,j}^{pred}. \quad (9)$$

Given the permuted encoder outputs, the grouping predictor conducts a binary classification task of whether d'_l is assigned to the same group as d'_{l-1}. Let the label at the position l be g_l. Then, we define g_l from G as follows:

$$g_l = \begin{cases} 1 & \text{if } G_{*,l} = G_{*,l-1} \text{ and } l > 1 \\ 0 & \text{else.} \end{cases} \quad (10)$$

The grouping loss is defined as follows:

$$\mathcal{L}_G = -\frac{1}{L} \sum_{l=1}^{L} \log p_l(g_l), \quad (11)$$

where p_l is the predicted probability distribution of the grouping predictor at position l.

Our final loss function is defined as the sum of the negative log-likelihood based translation loss $\mathcal{L}_T$ and alignment loss $\mathcal{L}_A$:

$$\mathcal{L} = \mathcal{L}_T + \mathcal{L}_A = \mathcal{L}_T + \alpha\mathcal{L}_D + \beta\mathcal{L}_P + \gamma\mathcal{L}_G, \quad (12)$$

where we set $\alpha = \beta = \gamma = 0.5$ for all the experiments.

4

3.2.3 Inference

During inference, Aligner sequentially predicts the duplication, permutation, and grouping matrices to compute the aligned decoder inputs d as depicted in Figure 1b. The duplication predictor in Aligner infers $\hat{c}_m$ at each position m; then, we can directly construct a duplication matrix $\hat{D}$ using Equation 5. The permutation predictor predicts the distribution of the target position P^{pred}. We obtain a permutation matrix $\hat{P}$ that minimizes the KL-divergence as follows:

$$\hat{P} = \arg\min_P(-\sum_i \sum_j P_{i,j} \log P_{i,j}^{pred}). \quad (13)$$

We utilize the *linear sum assignment problem solver* provided by Jones et al. (2001) to find $\hat{P}$. The grouping predictor infers the binary predictions $\hat{g}_l$ from the permuted encoder outputs. We construct a grouping matrix $\hat{G}$ using $\hat{g}_l$ and Equations 6 and 10. With a predicted alignment matrix $\hat{A} = \hat{G} \cdot \hat{P} \cdot \hat{D}$, Aligner constructs the decoder inputs using Equation 4, and the decoder performs translation from the aligned inputs.

3.2.4 Decoding Strategies

For the re-scoring based decoding method, we select candidates of alignments using the predicted distributions in the duplication and grouping predictors.

We identify m' positions in the outputs of the duplication predictor, where the probability of the predicted class is low. We then construct a $2^{m'}$-candidate pool where the predictions in part of the m' positions are replaced with the second probable class. Next, we identify the top-a candidates with the highest joint probabilities. Similarly, we construct a $2^{l'}$-candidate pool and identify b candidates in the grouping predictor for the a candidates. Finally, we rank $a \cdot b$ translations for the alignments candidates using a teacher ART model and select the best translation among them.

3.3 Architecture of AligNART

We use the deep-shallow (12-1 for short) Transformer (Vaswani et al., 2017) architecture (i.e., 12-layer encoder and 1-layer decoder) proposed by Kasai et al. (2020b) for two reasons. First, a deeper encoder assists Aligner to increase the estimation accuracy of the alignment matrix during inference. Second, the deep-shallow architecture improves the inference speed since the encoder layer has no cross-attention module compared to the decoder

layer. The architecture of the duplication, permutation, and grouping predictor is shown in the Appendix.

3.4 Alignment Score Filtering

Some alignment tools such as GIZA++ (Och and Ney, 2003) provide an alignment score for each sentence pair as a default. Samples with low alignment scores are more likely to contain noise caused by sentence pairs or alignment tools. For GIZA++, we filter out a fixed portion of samples with low alignment scores to ease the alignment estimation. Since the pair of long sentences tends to be aligned with a low score, we apply the same filtering portion for each target sentence length.

4 Experimental Setups

4.1 Datasets and Preprocessing

We evaluate our method on two translation datasets: WMT14 English-German (En-De) and WMT16 English-Romanian (En-Ro). WMT14 En-De/WMT16 En-Ro datasets contain 4.5M/610K training pairs, respectively.

For WMT14 En-De dataset, we use preprocessing pipelines provided by *fairseq*[1] (Ott et al., 2019). For WMT16 En-Ro dataset, we use the preprocessed corpus provided by Lee et al. (2018). Preprocessed datasets share a vocabulary dictionary between the source and target languages. We use *fast align* (FA) (Dyer et al., 2013) and GIZA++ (GZ), which is known to be more accurate than *fast align*, as word alignment tools. All the corpus are passed to the alignment tools at the subword-level. We filter out samples where the maximum number of duplications exceed 16. We explain the details of the alignment processing in the Appendix.

We use the sequence-level knowledge distillation method (KD) for the distillation set. Transformer ART models are trained to generate the distillation set for each translation direction.

4.2 Models and Baselines

We compare our model with several non-iterative NART baselines, and divide the non-iterative NART models into two types as aforementioned: *implicit dependency modeling* and *explicit modality reduction* (see Table 1). We also train the ART models and deep-shallow NAT for the analysis. Our models are implemented based on *fairseq*.

---

[1] https://github.com/pytorch/fairseq

Models	WMT14 En-De				WMT16 En-Ro			
	En→	De→	Time	Speedup	En→	Ro→	Time	Speedup
Autoregressive Models								
Transformer (Vaswani et al., 2017)	27.3	-	-	-	-	-	-	-
Transformer (ours)	27.4	31.4	314	×1.0	34.1	33.9	307	×1.0
Non-iterative Non-autoregressive Models (implicit dependency modeling)								
FlowSeq (Ma et al., 2019)	21.5	26.2	-	-	29.3	30.4	-	-
AXE (Ghazvininejad et al., 2020)	23.5	27.9	-	-	30.8	31.5	-	-
NAT-EM (Sun and Yang, 2020)	24.5	27.9	24	×16.4	-	-	-	-
NARLVM (Lee et al., 2020)	25.7	-	19	×15.0	-	28.4	18	×34.0
GLAT (Qian et al., 2021)	25.2	29.8	-	×15.3	31.2	32.0	-	×15.3
Imputer (Saharia et al., 2020)	25.8	28.4	-	×18.6	32.3	31.7	-	-
CTC (Gu and Kong, 2021)	**26.5**	**30.5**	-	×16.8	**33.4**	**34.1**	-	×16.8
Non-iterative Non-autoregressive Models (explicit modality reduction)								
NAT-FT (Gu et al., 2018)	17.7	21.5	39	×15.6	27.3	29.1	39	×15.6
Distortion (Zhou et al., 2020b)	22.7	-	-	-	29.1	-	-	-
ReorderNAT (Ran et al., 2021)	22.8	27.3	-	×16.1	29.3	29.5	-	×16.1
SNAT (Liu et al., 2021)	24.6	28.4	27	×22.6	**32.9**	32.2	27	×22.6
AligNART (FA, ours)	25.7	29.1	23	×13.6	31.7	32.2	22	×13.9
AligNART (GZ, ours)	**26.4**	**30.4**	24	×13.4	32.5	**33.1**	24	×13.0

Table 1: BLEU scores and inference speed of baselines and our model on four translation tasks. *Time* is an average sentence-wise latency in milliseconds. *Speedup* is a relative speedup ratio compared to the Transformer-based ART model with beam width 5.

AlignART is implemented based on the deep-shallow Transformer architecture. We set d_{model}/d_{hidden} to 512/2048 and the dropout rate to 0.3. The number of heads in multi-head attention modules is 8 except for the last attention module of the permutation predictor which is 1. We set the batch size to approximately 64K tokens for all the models we implement. All these models we implement are trained for 300K/50K steps on En-De/En-Ro datasets, respectively. For AlignART, we average 5 checkpoints with the highest validation BLEU scores in the 20 latest checkpoints.

For optimization, we use Adam optimizer (Kingma and Ba, 2015) with $\beta = (0.9, 0.98)$ and $\epsilon = 10^{-8}$. The learning rate scheduling follows that of Vaswani et al. (2017), starting from 10^{-7} and warms up to $5e$-4 in 10K steps. We use the label smoothing technique with $\epsilon_{ls} = 0.1$ for the target token distribution and each row of permutation matrix. The translation latency is measured on an NVIDIA Tesla V100 GPU.

5 Results

5.1 Main Results

Table 1 shows the BLEU scores, translation latency and speedup on WMT14 En-De and WMT16 En-

Models	WMT14 En-De		WMT16 En-Ro	
	En→	De→	En→	Ro→
FlowSeq (n=15)	23.1	28.1	31.4	32.1
NAT-EM (n=9)	25.8	29.3	-	-
GLAT (n=7)	26.6	**31.0**	32.9	33.5
ReorderNAT (n=7)	24.7	29.1	31.2	31.4
SNAT (n=9)	26.9	30.1	**34.9**	33.1
AlignART (FA, n=8)	26.5	30.3	32.7	33.1
AlignART (GZ, n=8)	**27.0**	**31.0**	33.0	**33.7**

Table 2: BLEU scores of non-iterative NART models with re-scoring decoding scheme of n candidates.

Ro. In explicit modality reduction, AlignART (FA) achieves higher BLEU scores than Distortion and ReorderNAT, which utilize the same alignment tool, since we leverage the entire alignment information rather than partial information such as fertility or ordering. Moreover, AlignART (GZ) significantly outperforms previous models for explicit modality reduction except for SNAT on En→Ro. In implicit dependency modeling, AlignART (GZ) outperforms Imputer and shows performance comparable to that of the state-of-the-art CTC-based model on En↔De by simply augmenting Aligner module to deep-shallow NAT. In this study, we focus on introducing complete information in word alignments; we do not modify the objective function,

	En→De			De→En		
Models	D	P	G	D	P	G
AligNART (FA, w/o KD)	0.76/0.85	0.55/0.74	0.96/0.98	0.77/0.84	0.59/0.74	0.96/0.98
AligNART (FA, w/ KD)	0.75/0.89	0.53/0.83	0.95/1.00	0.76/0.88	0.57/0.84	0.96/1.00
AligNART (GZ, w/o KD)	0.69/0.78	0.76/0.91	1.00/1.00	0.71/0.82	0.81/0.92	1.00/1.00
AligNART (GZ, w/ KD)	0.66/0.88	0.71/0.94	1.00/1.00	0.68/0.88	0.76/0.95	1.00/1.00

Table 3: Duplication (D), permutation (P), and grouping (G) accuracy of Aligner on WMT14 En-De validation set. Accuracy on raw and distilled datasets are written on the left and right of slash, respectively.

Source	Denken Sie , dass die Medien zu viel vom PS_ G erwarten ?
Reference	Do you think the media expect too much of PS_ G ?
NAT (12-1)	Do you think that the expect expect much from the PS_ G ?
Ours — Duplication	Denken Denken Sie , dass die Medien zu viel vom vom PSG PSG erwarten ?
Ours — Permutation	Denken Sie Denken , dass die Medien erwarten zu viel vom vom PSG PSG ?
Ours — Grouping	Denken Sie Denken , dass die Medien erwarten zu viel vom vom PSG PSG ?
Ours — Output	Do you think that the media expect too much from the PS_ G ?

Table 4: Visualized translation example of deep-shallow NAT and AligNART (FA) on WMT14 De→En validation set. "_" stands for subword tokenization. Highlighted tokens in duplication, permutation, and grouping processes are modified by the each module of Aligner. Highlighted tokens in output correspond to the tokens highlighted with the same colors in the previous processes. Note that Aligner first applies mean pooling to convert subword-level encoder outputs into word-level, as explained in the Appendix.

	fast align		GIZA++	
Models	En→	De→	En→	De→
AligNART	25.7	29.1	26.4	30.4
- Infer with D=I	15.5	18.1	11.5	15.2
- Infer with P=I	19.4	22.2	21.5	24.7
- Infer with G=I	21.9	27.1	26.4	30.4

Table 5: BLEU scores of Aligner ablation study on WMT14 En-De test set.

which can be explored in the future work.

Table 2 shows the BLEU scores with re-scoring decoding strategies of the non-iterative NART models. We set $m' = l' = 4$, $a = 4$, and $b = 2$ for 8 candidates. AligNART outperforms the baselines on En→De and Ro→En, and shows performance similar to that of GLAT on De→En. In non-iterative NART for explicit modality reduction, AligNART shows the best performance on En↔De and Ro→En.

5.2 Analysis of Aligner Components

In this section, we investigate the accuracy, example, and ablation results of Aligner components as shown in Table 3, 4, and 5, respectively. Note that we partially provide the ground truth D or P matrices during the accuracy measurement.

Knowledge Distillation In Table 3, a comparison of accuracy between raw and distilled datasets shows that KD significantly decreases *multi-modality* of each component. After KD, AligNART shows marginally reduced accuracy on the raw dataset, but high prediction accuracy in each component on the distillation set, resulting in increased BLEU scores.

Alignment Tool Before KD, AligNART using *fast align* and GIZA++ have accuracy bottlenecks in permutation and duplication predictors, respectively, as shown in Table 3. The results imply that the alignment tools have different degrees of *multi-modality* on the D, P, and G matrices, which can be explored in the future work.

Qualitative Study Table 4 shows an example of addressing the *multi-modality problem*. Deep-shallow NAT monotonically copies the encoder outputs and suffers from repetition and omission problems. AligNART (FA) does not show the inconsistency problems thanks to the well-aligned decoder inputs, which significantly reduces the modality of the target distribution. We also conducted a case study on predicted alignments and their translations during re-scoring as shown in the Appendix.

Ablation Study We conduct an analysis of alignment estimation by ablating one of the predictors

	WMT14 En-De	
	En→	De→
FlowSeq (w/o KD)	18.6	23.4
AXE (w/o KD)	20.4	**24.9**
Imputer (CTC, w/o KD)	15.6	-
CTC (w/o KD)	18.2	-
NAT (12-1, w/o KD)	8.5	13.3
NAT (12-1, w/ KD)	18.9	23.4
AligNART (FA, w/o KD)	**20.7**	24.0
AligNART (GZ, w/o KD)	18.3	23.2

Table 6: BLEU scores of non-iterative NART models on WMT14 En-De test set, with or without KD.

	WMT14 En-De	
	En→	De→
Gold test set	0.04%	0.03%
CMLM-base (5 iterations)	0.72%	-
AXE	1.41%	1.03%
Imputer (CTC)	0.17%	0.23%
GLAT	1.19%	1.05%
NAT (12-1, w/o KD)	33.94%	27.78%
NAT (12-1, w/ KD)	11.83%	9.09%
AligNART (GZ, w/o KD)	0.76%	1.33%
AligNART (GZ, w/ KD)	0.33%	0.33%

Table 7: Token repetition ratio of NART models on WMT14 En-De test set.

	WMT14 En-De	
	En→	De→
NAT (12-1)	18.9	23.4
- Cross attention	17.2	21.9
AligNART (GZ)	26.4	30.4
- Score filtering	26.2	30.0
- Cross attention	26.1	29.9
- 12-1 architecture	24.9	29.1

Table 8: Ablation results of deep-shallow NAT and AligNART (GZ) on WMT14 En-De test set.

during inference. We ablate each module in Aligner by replacing the predicted matrix with an identical matrix I. The results in Table 5 indicate that each module in Aligner properly estimates the decomposed information in word alignments. However, there is an exception in GIZA++ where many-to-one mapping does not exist, resulting in performance equal to that without the grouping predictor. We observe that AligNART achieves BLEU scores comparable to those of CTC-based models on En↔De even with the ground truth word alignments of partial information.

5.3 Analysis of Modality Reduction Effects

To evaluate the modality reduction effects of Alig-NART, we conducted experiments on two aspects: BLEU score and token repetition ratio. Table 6 shows the BLEU scores on WMT14 En-De. For En→De, AligNART using *fast align* without KD achieves higher BLEU scores than previous models without KD and deep-shallow NAT with KD. The results indicate that our method is effective even without KD, which is known to decrease data complexity (Zhou et al., 2020a). On the other hand, alignments from GIZA++ without KD are more complex for AligNART to learn, resulting in lower BLEU scores than deep-shallow NAT with KD.

Ghazvininejad et al. (2020) measured the token repetition ratio as a proxy for measuring *multi-modality*. The token repetition ratio represents the degree of the inconsistency problem. In Table 7, the token repetition ratio of AligNART is less than that of the CMLM-base (Ghazvininejad et al., 2019) of 5 iterations, AXE, and GLAT. We also observe that the decline in the token repetition ratio from Aligner is significantly larger than that from KD. Combined with the results from Table 6, alignment

information adequately alleviates the token repetition issue even in the case where the BLEU score is lower than that of deep-shallow NAT with KD.

5.4 Ablation Study

We conduct several extensive experiments to analyze our method further as shown in Table 8 and 9. Each of our method consistently improves the performance of AligNART.

Cross Attention As shown in Table 8, we ablate the cross attention module in the decoder to observe the relationship between aligned decoder inputs and alignment learning of the cross attention module. We train AligNART and deep-shallow NAT without a cross attention module for comparison. AligNART without the cross attention module has a smaller impact on the BLEU score than the deep-shallow NAT. The cross attention module is known to learn alignments between source and target tokens (Bahdanau et al., 2015), and the result implies that aligned decoder inputs significantly offload the role of the cross attention module.

Deep-shallow Architecture Deep-shallow architecture heavily affects the BLEU scores of Alig-

	0%	1%	5%	10%	20%
En→	26.2	26.1	26.4	26.2	26.2
De→	30.0	30.2	30.4	30.4	30.1

Table 9: Alignment score filtering ratio and BLEU scores on WMT14 En-De test set.

NART as shown in Table 8. The results indicate that the deep encoder assists alignment estimation, whereas the shallow decoder with aligned inputs has a lower impact on performance degeneration.

Alignment Score Filtering We investigate the trade-off between the alignment score filtering ratio and BLEU score using AligNART (GZ) presented in Table 9. Samples with low alignment scores are more likely to contain noise caused by distilled targets or an alignment tool. We observe that filtering out of 5% of the samples improves the BLEU score in both the directions. Surprisingly, increasing the filtering ratio up to 20% preserves the performance thanks to the noise filtering capability.

6   Related Work

6.1   Non-iterative NART

After Gu et al. (2018) proposed NAT, non-iterative NART has been investigated in various directions to maximize translation speed while maintaining translation quality. Shao et al. (2019), Shao et al. (2020), and Ghazvininejad et al. (2020) address the limitations of conventional cross entropy based objectives that overly penalize consistent predictions. Lee et al. (2018), Ma et al. (2019), Shu et al. (2020), and Lee et al. (2020) introduce latent variables to model the complex dependencies between target tokens. Saharia et al. (2020) and Gu and Kong (2021) apply CTC loss to the NMT domain. Qian et al. (2021) provide target tokens to the decoder during training using the glancing sampling technique.

6.2   Alignment in Parallel Generative Models

In other domains, such as text-to-speech (Ren et al., 2019; Kim et al., 2020; Donahue et al., 2020), a common assumption is a monotonicity in the alignments between text and speech. Given this assumption, only a duration predictor is required to alleviate the length-mismatch problem between text and speech. On the other hand, modeling the alignment in the NMT domain is challenging since the alignment contains additional ordering and grouping information. Our method estimates an arbitrary alignment matrix using *alignment decomposition*.

6.3   Improving NMT with Enhanced Information

To alleviate the *multi-modality problem* of NART models, Gu et al. (2018), Akoury et al. (2019), Zhou et al. (2020b), Ran et al. (2021), and Liu et al. (2021) provide additional sentence information to the decoder.

Alignment is considered as a major factor in machine translation (Li et al., 2007; Zhang et al., 2017). Alkhouli et al. (2018) decompose the ART model into alignment and lexical models. Song et al. (2020) use the predicted alignment in ART models to constrain vocabulary candidates during decoding. However, the alignment estimation in NART is much challenging since the information of decoding outputs is limited. In NART, Gu et al. (2018), Zhou et al. (2020b), and Ran et al. (2021) exploit partial information from the ground truth alignments. In contrast, we propose the *alignment decomposition* method for effective alignment estimation in NART where we leverage the complete alignment information.

7   Conclusion and Future Work

In this study, we leverage full alignment information to directly reduce the degree of the *multi-modality* in non-iterative NART and propose an *alignment decomposition* method for alignment estimation. AligNART with GIZA++ shows performance comparable to that of the recent CTC-based implicit dependency modeling approach on WMT14 En-De and modality reduction capability. However, we observe that AligNART depends on the quality of the ground truth word alignments, which can be studied in the future work. Furthermore, we can study on the combination of AligNART and implicit dependency modeling methods.

References

Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. 2019. Syntactically supervised transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1281, Florence, Italy. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium. Association for Computational Linguistics.

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

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Jeff Donahue, Sander Dieleman, Mikolaj Binkowski, Erich Elsen, and Karen Simonyan. 2020. End-to-end adversarial text-to-speech. 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, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

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.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, Online. Association for Computational Linguistics.

Junliang Guo, Linli Xu, and Enhong Chen. 2020. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 376–385, Online. Association for Computational Linguistics.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001. SciPy: Open source scientific tools for Python.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020a. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2020b. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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 *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Jason Lee, Raphael Shu, and Kyunghyun Cho. 2020. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1006–1015, Online. Association for Computational Linguistics.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 720–727, Prague, Czech Republic. Association for Computational Linguistics.

Ye Liu, Yao Wan, Jianguo Zhang, Wenting Zhao, and Philip Yu. 2021. Enriching non-autoregressive transformer with syntactic and semantic structures for neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1235–1244, Online. 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.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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. 2021. Guiding non-autoregressive neural machine translation decoding with reordering information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13727–13735.

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. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3165–3174.

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.

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, 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, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8846–8853.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *AAAI*, pages 8886–8893.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhi-Hong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3011–3020.

Zhiqing Sun and Yiming Yang. 2020. An EM approach to non-autoregressive conditional sequence genera-

tion. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1524–1534, Vancouver, Canada. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020a. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Long Zhou, Jiajun Zhang, Yang Zhao, and Chengqing Zong. 2020b. Non-autoregressive neural machine translation with distortion model. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 403–415. Springer.

Appendix

A Mappings in Alignment

In general, there are one-to-one, one-to-many, many-to-one, and many-to-many mappings excluding zero-fertility and spurious word cases (see Figure 2). Distortion and ReorderNAT cannot represent many-to-one, many-to-many, and spurious word cases. The grouping predictor in AligNART models many-to-one and many-to-many mappings. The addition of a *spurious* token, which is applied to AligNART (FA), enables us to address the spurious word case, which is explained in Section C.2. During the experiments, we observe that the introduction of a *spurious* token degrades the performance for GIZA++. We guess the reason of the degradation is that alignment matrix from GIZA++ contains more than two times as many empty rows as that of *fast align* on WMT14 En-De.

B Architecture of Aligner

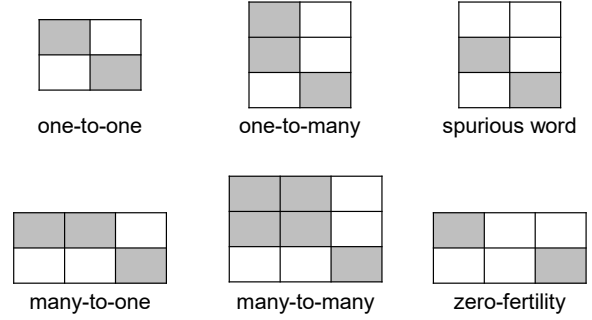The duplication predictor and grouping predictor modules consist of a convolutional layer, ReLU ac-

Figure 2: Types of mapping in word alignments. Row and colum correspond to the target and source tokens, respectively.

tivation, layer normalization, dropout, and a projection layer, same as the phoneme duration predictor in FastSpeech (Ren et al., 2019), which is a parallel text-to-speech model.

The permutation predictor in Aligner consists of three encoder layers: pre-network, query/key network, and single-head attention module for the outputs. Note that the outputs of the pre-network are passed to the query and key networks. To prevent the predicted permutation matrix from being an identity matrix, we apply a gate function to the last attention module in the permutation predictor to modulate the probabilities of *un-permuted* and *permuted* cases. We formulate the output of gated attention as follows:

$$g = \sigma(Q \cdot u) \tag{14}$$

$$\bar{P}^{pred} = softmax(M + QK^T) \tag{15}$$

$$P^{pred} = D_g + (I - D_g) \cdot \bar{P}^{pred}, \tag{16}$$

where σ is the sigmoid function and Q/K is the output of the query/key network, respectively. g is the probability of an *un-permuted* case. M is a diagonal mask matrix, where the values of the diagonal elements are $-inf$. I is an identical matrix and D_g is a diagonal matrix with g as the main diagonal.

C Alignment Processing

C.1 Word-to-subword Alignment

To reduce the complexity of alignment, we further assume that the alignment process is conducted at the word-level. We decompose the alignment matrix into the source subword to source word matrix S and the source word to target subword matrix A^{ws} as depicted in Figure 3. Since S is always given, A^{ws} is the only target to be learned. First,
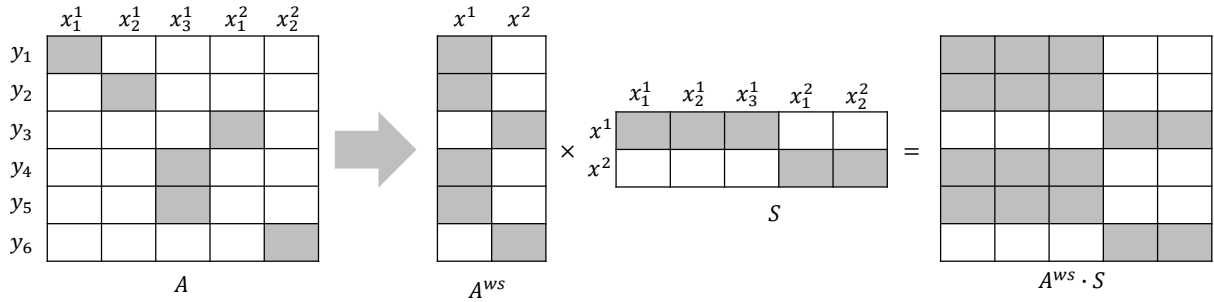
Figure 3: Example of word-to-subword matrix decomposition technique. Row and column correspond to input and output tokens, respectively. y_i denotes the i-th subword of the target sentence. x^i denotes the i-th word of the source sentence and x^i_j denotes the j-th subword of the i-th word of the source sentence.

we derive the source subword to target subword matrix A using the alignment tool. A^{ws} is achieved by clipping the maximum value of $A \cdot S^\top$ to 1. A^{ws} reduces the search space because of the assumption that source tokens duplicate, permute, and group at the word-level. However, there is a trade-off between the simplicity and resolution of information. The recovered source subword to target subword matrix $A^{ws} \cdot S$ loses the subword-level information as shown in the rightmost matrix in Figure 3.

C.2 Filling Null Rows in Alignment Matrix

The output of the alignment tool usually contains empty rows which means that no aligned source token exists for certain target tokens. We select two strategies to fill the null rows: (i) copy the alignment from the previous target token, or (ii) introduce a special *spurious* token. For the second strategy, we concatenate a special *spurious* token at the end of the source sentence. If the current and previous target tokens belong to the same word, we follow (i). The remaining target tokens of the null alignment are aligned to the *spurious* token.

C.3 Details of Alignment Tool Configuration

For *fast align*, we follow the default setting for forward/backward directions and obtain symmetrized alignment with the *grow-diag-final-and* option. We apply the word-to-subword alignment technique and *spurious* token strategy for null alignments. For GIZA++, we apply the word-to-subword alignment technique and copy the alignment from the previous target token for null alignment. We set the alignment score filtering ratio to 5%.

D Case Study

To analyze various alignments and their translations during re-scoring decoding, we conduct a case study on WMT14 De→En validation set as shown in Figure 4. The two translations have different orderings: *the telescope's tasks* and *the tasks of the telescope*. In this sample, we observe that AlignNART (i) can capture non-diagonal alignments, (ii) models multiple alignments, and (iii) translates corresponding to the given alignments.
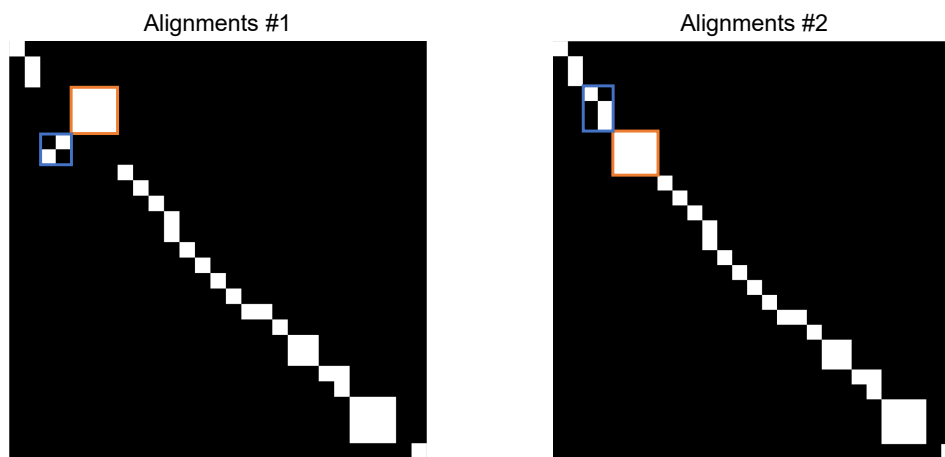
Source	Eine der Aufgaben des Tel_ esk_ ops : Es soll nach Licht von den ersten Ster_ nen und Galax_ ien nach dem Ur_ kn_ all suchen .
Reference	One of the tel_ esc_ ope 's tasks is to search for light from the first stars and galax_ ies that emerged after the Big B_ ang .
Alignments #1	One of the tel_ esc_ ope 's tasks : it should search for light from the first stars and galax_ ies after the Big B_ ang .
Alignments #2	One of the tasks of the tel_ esc_ ope : it should search for light from the first stars and galax_ ies after the Big B_ ang .

Figure 4: Translation and alignment estimation example on WMT14 De→En validation set. Tokens matched to the alignment matrix have same colors (blue and orange). The special token "_" stands for the subword tokenization.