

A Survey of Non-Autoregressive Neural Machine Translation

Feng Li ¹ , Jingxian Chen ¹ and Xuejun Zhang ^{1,2,3,*} ¹ School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China² Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning 530004, China³ Guangxi Big White & Little Black Robots Co., Ltd., Nanning 530007, China

* Correspondence: xjzhang@gxu.edu.cn

Abstract: Non-autoregressive neural machine translation (NAMT) has received increasing attention recently in virtue of its promising acceleration paradigm for fast decoding. However, these splendid speedup gains are at the cost of accuracy, in comparison to its autoregressive counterpart. To close this performance gap, many studies have been conducted for achieving a better quality and speed trade-off. In this paper, we survey the NAMT domain from two new perspectives, i.e., target dependency management and training strategies arrangement. Proposed approaches are elaborated at length, involving five model categories. We then collect extensive experimental data to present abundant graphs for quantitative evaluation and qualitative comparison according to the reported translation performance. Based on that, a comprehensive performance analysis is provided. Further inspection is conducted for two salient problems: target sentence length prediction and sequence-level knowledge distillation. Accumulative reinvestigation of translation quality and speedup demonstrates that non-autoregressive decoding may not run fast as it seems and still lacks authentic surpassing for accuracy. We finally prospect potential work from inner and outer facets and call for more practical and warrantable studies for the future.

Keywords: non-autoregressive; autoregressive; machine translation; fast decoding



Citation: Li, F.; Chen, J.; Zhang, X. A Survey of Non-Autoregressive Neural Machine Translation. *Electronics* **2023**, *12*, 2980. <https://doi.org/10.3390/electronics12132980>

Academic Editors: Álvaro Carrera Barroso, Oscar Araque, Lorenzo Gatti and Kyriaki Kalimeri

Received: 3 June 2023

Revised: 27 June 2023

Accepted: 29 June 2023

Published: 6 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Developing fast and accurate machine translation in a wide range of applications is an elementary orientation for both research and industry. This decade has witnessed the rapid progress of neural machine translation (NMT) benefiting from the application of various artificial neural networks [1–6]. Particularly, Transformer has realized state-of-the-art performance and has become the de facto mainstream architecture [6–10]. The vanilla Transformer adopts an encoder–decoder framework [1,2]. Taking a sentence from one language as input on the source side, the framework converts it into another language as an output on the target side. Among this, the encoder maps the input sentence into hidden representation, and then the decoder decodes the hidden representation into an output. While this process can be trained with high parallelism via teacher forcing [11], the decoding of the inference stage is autoregressive and non-parallel. In order to produce a current token, the previous time step predication must be the extra decoder input, which means that each token is predicted not only based on the source input but also the previously generated token, sequentially forming a left-to-right and word-by-word generating arrangement. Although it achieves impressive performance success, this intrinsically sequential Autoregressive Transformer (AT) process cannot be parallelizable during inference, leading to a high inference latency and preventing industrial application, in which low latencies and simultaneous responses are demanded.

To mitigate this, a flurry of recent work has been developed for fast decoding towards non-autoregressive neural machine translation (NAMT) (Figure 1). Gu et al. [12] firstly proposed a Non-Autoregressive Transformer (NAT) based on the Transformer network, which could generate each token independently and simultaneously. The NAT assumed that

the prediction of each token on the target side was conditionally independent and merely based on a source input without respecting previously generated words, thus generating all target tokens in parallel and at once. Consequently, NAT obtained a latency of 39 ms per sentence and 15.6 times speedup of decoding speed, compared to the 607 ms per sentence of Transformer. However, the great speedup gains were at the cost of accuracy, which could have been up to 5.76 points compared to its counterpart (i.e., Autoregressive Transformer), according to the BLEU score [13], a general criterion for automatically evaluating the quality of machine translation.

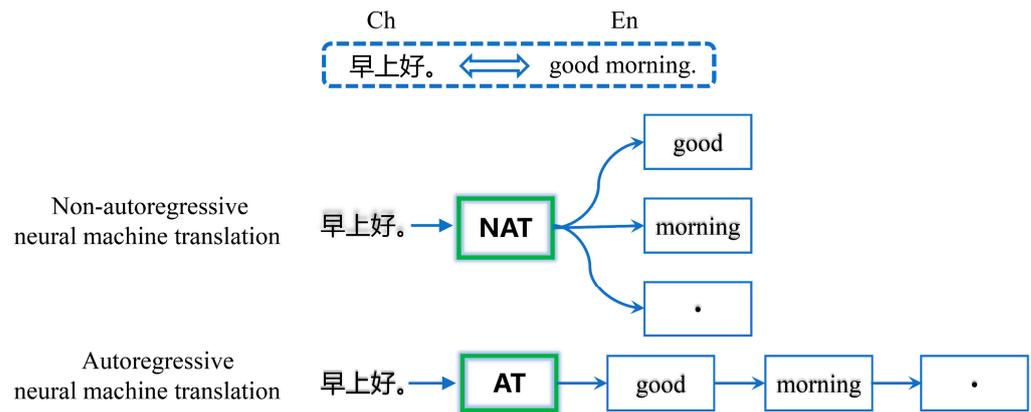


Figure 1. An instance for non-autoregressive neural machine translation (NAMT) and autoregressive neural machine translation (AMT). The Chinese term of “早上好” can be accurately translated into the English term of “Good morning” in this figure.

This accuracy drop distinctly stems from the strongly conditional independence assumption on the target side, which, in the meantime, induces the prominent challenge of capturing the highly multimodal distribution of target translations (i.e., multiple feasible translations correspond to a single source). Two prevailing problems empirically observed in the non-autoregressive output are: (1) Over-translated data, where repeated words are generated at in consecutive positions. For example, given two German source sentences “Danke schön” and “Vielen Dank”, which can be translated into “thank you” and “thank you very much” in English, respectively, NAT could generate “thank” for the second output token, even though the word “thanks” has already been selected, and it could output possible translations such as “thanks thank” and “thanks very very much”. (2) Under-translated data, where the semantics of several phrases from the source can be blundered or missed in the output. See a specific instance in Table 1 for more intuitional detail regarding each phenomenon.

Table 1. Over-translated and under-translated phenomena of NAMT.

Vielen Dank		
Reference	Thank you very much	(standard translation)
Over-translated	Thank you much much	(repeated translation)
Under-translated (1)	Thanks you very much	(mistaken semantic)
Under-translated (2)	Thank very much	(missing semantic)

Both of these issues suggest the inferior ability of NAT to capture the multimodal distribution of output space compared to AT. More precisely, to maximize the likelihood of an entire output sentence, AT selects words with maximal probability for each token. This searching process is effectively performed via beam search based on the previous tokens as a certain restriction, with narrowed solution space of tractable width and depth. For example, given the source sentence “谢谢你” in Chinese and target translation “thank you” in English, AT is unlikely to produce any other words in addition to “you” when the

previously generated word is “thank”. By contrast, the prior restriction disappears in NAT as the sequential dependencies between tokens on target side are removed. There could be multiple alternative words mapped to an unpredicted token in the output of NAMT, resulting in an enlarged solution space that is intractable for NAT. Consequently, in order to minimize the loss, NAT seeks to select the word with highest likelihood for each target token in spite of the correlations between them.

To close the accuracy gap between the AT and NAT models while alleviating the inconsistency in the generation of NAMT, successive endeavors have been made by researchers. A line of work aims to loosen the dependencies on the target side by utilizing latent variables or other intermediates as pivots to accomplish the transformation from source to target, realizing fully parallel non-autoregressive generation. Another branch of this study focuses on reconstructing the interdependencies through multiple-step decoding and iteratively refining the generations until the final output. Except for managing the sequential dependency on the target side, some researchers turn to implementing alternative training strategies, including new objectives and multiple training methods, considerably boosting the performance of NAMT.

In this paper, we provide an elaborate review of non-autoregressive neural machine translation during the past 5 years. It is worth mentioning that a preprinting study [14] is similar to our work, which describes this domain from four aspects concurrently, covering other non-autoregressive generation tasks. However, different from it, we center more on neural machine translation and inspect this field with decent insight from two new perspectives: dependency management on the target side and training arrangement for NAT models. Other than delivering quantitative description and qualitative comparison for various methods, we also anticipate promising future directions for this area, following up the latest findings, including simultaneous translation [15,16], automatic speech recognition [17–19], speech translation [20–22], image caption [23], and text editing [24–29], as well as the emerging large language models [30,31].

Our contribution to this community can be summarized in three aspects:

1. We provide a concise retrospective on the technology evolving of non-autoregressive neural machine translation from the different viewpoints of target-side dependency management and training strategy arrangement.
2. We made a comprehensive comparison among the methods applied in this field according to both effectiveness and accuracy via quantitative evaluation of the reported data and qualitative analysis based on the proposed theory.
3. In addition to the review, the practices for fast decoding in corresponding tasks are also described, along with the challenges of NAMT and the prospects for future direction in this area.

The rest of this paper is organized as follows: Section 2 briefly introduces the preliminary knowledge of machine translation; Section 3 elaborates various efforts made to promote the performance of NAMT; Section 4 provides a quantitative comparison and a qualitative analysis for the methods mentioned before; in Section 5, crucial problems of NAT models in common are inspected; further critical discussion and future work are conveyed in Section 6; and the final Section 7 concludes the whole paper.

2. Preliminary

In this section, we first introduce some fundamental knowledge about translation. A brief description of the comparison between human translation (HT) and machine translation (MT) is also provided. Then, we pass the preliminary statements of the AMT with Transformer. Finally, the depiction of the principle theory in NAMT is detailed, along with the problems that need to be grappled with.

Translation is the creation of a translated text to perform a function. The connection it maintains with its source text will be materialized according to the functions expected or required by the translation. Currently, translation methods can be divided into human

translation and machine translation according to whether the translation is performed by humans or machines.

Machine translation generally refers to the conversion of one natural language into another natural language by a machine. Literally, the translated texts from machine translation are considerably decent in some specific scenarios. In more open fields, nevertheless, it does not perform desirably. Specifically, in simultaneous translation, the quality and accuracy of machine translation need further advancements to meet practical applications. For the translation of novels, machine translation still lags behind manual work. It is also more entertaining than a realistic deployment when translating poems via machine. In comparison, human translation has certain strengths over machine translation in simultaneous interpretation, literary translation, and other aspects of high translation quality requirements. Machine translation has viable advantages over human translation for large amounts of translation, such as web page translation and document translation.

(a) Machine translation

The notation of machine translation, formally proposed in 1949 by Weaver [32], is expected to automatically transfer sequences from one language to another. Following the technology of computational machines, researchers at Georgetown University attempted the first automatic machine translation in 1954, opening the study of syntactic-driven machine translation systems, which perform translation via manually formulated syntactic rules. However, the world's thousands of languages change rapidly over time and vary dramatically in regions. In spite of the high overheads of various human-crafted syntactic rules, this schema suffers from limited coverage and poor robustness. At the beginning of the 1980s, the electronization of literal resources enriched the available data and linguistic corpus for language learning, leading to the emergence of data-driven machine translation. The series of models formulates the transformation between languages based on the theory of statistical mathematics (i.e., statistical machine translation [33,34]). On top of this, since 2013, the boom of deep learning techniques has brought about another boost for MT. By using various artificial neural networks, neural machine translation realizes state-of-the-art accuracy via an end-to-end framework without the many necessary preprocessing steps that statistical machine translation needs.

(b) Autoregressive neural machine translation

Most autoregressive neural machine translation (AMT) models attaining SOTA performance on the task of MT use the Transformer architecture [6]. The vanilla Transformer adopts an encoder–decoder framework. Generally, the encoder is composed of six identical sub-layers, which contain a multi-head self-attention module and a feed-forward neural network. The decoder is also a stack of the same six sub-layers. In addition to the two sub-modules aforementioned in each encoder's sub-layer, a different layer, namely, masked multi-head attention, is inserted on top of encoder outputs. Considering a source input sentence $X = \{x_1, x_2, \dots, x_M\}$ and the corresponding target sentence $Y = \{y_1, y_2, \dots, y_N\}$, Transformer models the translation from X to Y as a probability conditioned on input X and previous predictions, which can be formulated as a chain of conditional probabilities:

$$f(X \rightarrow Y) = P(Y|X) = P(y_1, y_2, \dots, y_N | X; \theta_{AT}) = \prod_{i=1}^N P(y_i | y_{<i}, X; \theta_{AT}) \quad (1)$$

in which y_i and $y_{<i}$ indicate the present i -th time step token and the previously generated tokens, respectively. N is the length of the entire target sentence, and θ_{AT} is the parameter of the model. In particular, the encoder first maps a source input sentence $X = \{x_1, x_2, \dots, x_M\}$ into a continuous hidden representation $Z = \{z_1, z_2, \dots, z_M\}$ and passes it to the decoder. The decoder then decodes the hidden representations and generates the target output $Y = \{y_1, y_2, \dots, y_N\}$ one element at a time. At each time step, the prediction is autoregressive by consuming previously predicted tokens as extra decoder inputs when predicting the next, leading to a sequential left-to-right and word-by-word generating paradigm.

This autoregressive decoding process is often trained to converge using the standard cross-entropy loss function, which minimizes the negative log-probability as follows:

$$\mathcal{L}_{mxe}(\theta) = - \sum_{i=1}^N \log(P(y_i|y_{<i}, X; \theta_{AT})) \quad (2)$$

In the training stage, this translation process mentioned above can be paralleled by employing the algorithm of teacher force by feeding the prepared ground-truth words to the model as target side inferences. However, the intrinsic sequential property that lies in autoregressive decoding is inevitable during the inference phase. It is unavailable to use ground-truth words as the target side, which, consequently, causes high inference latency and hinders the industrial application, especially in scenarios where low latency or real-time responding is demanded. In addition, the effectiveness inconsistency during decoding between training and inference limits the transformer's parallelism and also hampers the full utilization of the parallel processing capacity of the graphical processing unit (GPU).

(c) Non-autoregressive neural machine translation

To speed up the autoregressive decoding, Gu et al. (2017) first proposed the Non-Autoregressive Transformer (NAT) to generate target tokens independently and simultaneously, namely, non-autoregressive neural machine translation (NAMT) [12]. The NAT adopts a similar encoder–decoder framework. However, different from AT, the NAT's decoder discards the dependencies among tokens on the target side, assuming that the prediction of every target token is independent from each other, and thus generates all tokens in parallel and at once. Given a source sentence $X = \{x_1, x_2, \dots, x_M\}$ and the corresponding target sentence $Y = \{y_1, y_2, \dots, y_N\}$, based on this naive assumption of conditional independence, NAT models the translation from X to Y as:

$$f(X \rightarrow Y) = P(Y|X) = P(y_1, y_2, \dots, y_N | X; \theta_{NAT}) = P(N|X; \theta_{NAT}) \prod_{i=1}^N P(y_i|X; \theta_{NAT}) \quad (3)$$

y_i refers to the i -th time step token, N is the length of the target sentence, and θ_{NAT} is the parameter of NAT. In this formulation, NAT produces each word on the target side merely based on source input sentences without recalling preceding predictions, achieving a speedup of 39 ms, 15.6 times beyond the 607 ms per sentence of AT. However, coarsely removing the internal dependency between words in the target sentence does not yield desirable accuracy. The impressive speedup gains come at the cost of potential performance degradation compared to its AT counterpart. Typically, NAT suffers from the multimodality problem that the model is poor at properly capturing the highly multimodal distribution of target side sentences since there are multiple feasible translations for a single source input, inducing translation inconsistency on two aspects, i.e., over-translated and under-translated. In particular, the over-translated problem refers to the phenomenon of repeated translations in which the same words are generated at adjacent token positions, e.g., with the English sentence "Thank you" which can be translated into any one of the three "Danke", "Danke schön", or "Vielen Dank" in German, the NAT could output possible translations such as "Danke Dank" and "Vielen schön". Another under-translated problem means the model fails to convert complete semantic information from the source side into the target side, causing words to be missing or lexical errors, leading to inferior translation quality.

3. Proposed Approaches

Given a source sentence, NAMT is intuitively akin to asking each member of a translation panel to offer one word for a particular position without interacting with each other. The final translation is formed by collecting every single member's answer independently. On the one hand, the mission is challenging because limited information can be acquired merely based on the source. On the other hand, the omitted communications among

experts amplify the difficulty in making a favorable translation decision. To contend with this, feasible tactics can be empirically formulated from two distinct inspirations based on the analogy. One is to manage the lost dependency, including reducing this dependency through resorting to other alternatives or recovering it as much as possible. Proposed models following this inspiration cover fully non-autoregressive decoding (FNAD) and iterative non-autoregressive decoding (INAD), which are identified according to their decoding manner and methodology notation. Another is to exert effective training on each expert inside the group via new training goals or instrumental training strategies. Proposed models stimulated by this purpose comprise new training objectives (NTB) and multiple training strategies (MTS), which are classified in terms of their training motivation. Some works that integrate multiple mechanisms from the aforementioned categories are identified as multi-mechanism integrated (MI).

To mitigate the aforementioned inconsistency and achieve the trade-off between decoding speed and generation quality, extensive work has been investigated by researchers for NAMT. In light of the selection criteria above, in this section, we elaborate on the efforts that have been made in two aspects: dependency management and training arrangements. The former contains fully non-autoregressive decoding (FNAD) and iterative non-autoregressive decoding (INAD), and the latter involves new training objectives and multiple training strategies, respectively. In addition to that, some multi-mechanism integrated methods are also introduced. Figure 2 depicts a global profile for all methods.

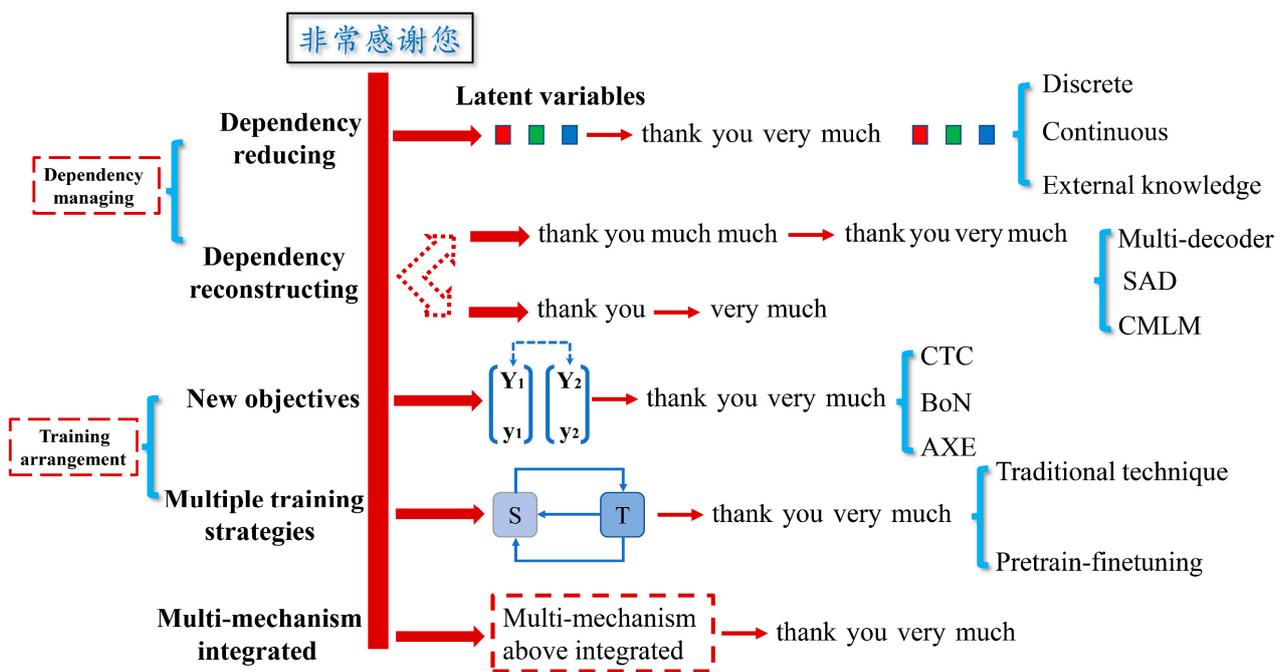


Figure 2. A global profile for all method categories.

3.1. Dependency Management

The strong independence assumption of NAMT is unnatural in reality, as there is inherent context among words in a sentence. Omitting the dependencies from the target side also results in a locally isolated prediction. Due to the lack of internal communication during inference, there may be repeated, missing, or wrong words submitted by experts from different positions. To remedy this, one distinct way is to relieve the dependencies on the target side by exploiting sequential information from the source side as much as possible, which refers to dependency reduction for fully non-autoregressive decoding (FNAD). Another natural solution is to gradually reconstruct the omitted dependencies on the target side, which involves dependency reconstruction via iterative non-autoregressive decoding (INAD).

3.1.1. Dependency Reducing for Fully NAD (FNAD)

Due to the fact that context dependencies on the target side are discarded, dependency reduction aims to relieve the dependence on the target side through modeling latent variables $z = (z_1, z_2, \dots, z_m)$ as a pivot to complete the conversion from source to target, thus generating all target tokens at once and in parallel. This category features non-iterative and one-pass decoding as well as fully non-autoregressive decoding (fully NAD). The modeling can be factorized into a chain of conditional probabilities based on x and z :

$$f(X \rightarrow Y) = P(Y|Z, X) = P(y_1, y_2 \dots y_N | Z, X; \theta_{NAT}) = P(Z|X; \theta_{NAT}) \prod_{i=1}^N P(y_i|Z, X; \theta_{NAT}) \quad (4)$$

Some studies use discrete latent variables. Specifically, Gu et al. [12] first used fertilities as discrete latent variables, which are produced by an exclusive fertility predictor, to copy the source sentence as the decoder input. The fertility value of each source token specifies how many times this word will be duplicated to form the decoder input. Additionally, the total sum of each fertility for a single input sentence equals the length of the corresponding target sentence. Kaiser et al. [35] then extended this work through the Latent Transformer. They used a transformer module to autoregressively construct discrete latent variable sequences with discretization techniques such as vector quantized autoencoders (VQ-VAE) [36] and improved semantic hashing [37]. Aurko Roy et al. [38] moved further by incorporating the EM algorithm to train VQ-VAE, which yielded significant improvements. Xuezhe Ma et al. [39] attempted a flow-based sequence-to-sequence model, a mathematical framework called generative flow, to approach the distribution of discrete latent variables. Jongyoon Song et al. [40] devised a specialized Aligner as an extra module to produce aligned decoder inputs. It helped the model progressively learn one-to-one mapping. Differently, DongNyeong Heo et al. [41] aimed to alleviate the defects of information redundancy, increased parameters, and semantic loss caused by an extra module of latent variable prediction.

Other works adopted different alternatives. One was to use syntactic knowledge as weak supervision. Nader Akoury et al. [42] replaced the discrete latent variables with syntactical parser chunks to simultaneously generate target sequences conditioned on autoregressively predicted constituency parsing sequences. Ye Liu et al. [43] integrated structured information, e.g., syntactic tags of Part-Of-Speech (POS) and semantic labels of Named-Entity-Recognition (NER), into latent variables as decoder inputs. Bao Yu et al. [44] instead substituted the syntactic labels with categorical codes that acted similar to fuzzy target categories for each target sequence without using syntactic trees. The other utilized positional information rather than syntactic knowledge. Ran Qiu et al. [45] modeled the reordering information of source inputs to guide the parallel decoding of NAT. Bao Yu et al. [46] explicitly modeled the positions of output tokens as latent variables for target side predictions.

3.1.2. Dependency Reconstructing for Iterative NAD (INAD)

Another distinct way to handle this deficiency is to accumulatively reconstruct the lost dependencies between target tokens, so called dependency reconstruction. This approach is characterized by multi-pass decoding as well as iterative non-autoregressive decoding (INAD).

The premier INAD method was proposed by Lee Jason et al. [47]. They treated the rebuilding phase as a conditional denoising process based on latent variables and deployed two non-autoregressive decoders to parallelly produce outputs through multiple passes. During inference, the outputs generated by the first decoder will be passed to the second decoder as inputs for iterative refinements until a prepended stopping criterion is met. Inspired by discrete latent variables and the iterative refinement method, Shu Raphael et al. [48] introduced continuous latent variables and executed refining in latent space rather than a discrete output space using a delta posterior. Similarly, Lee Jason et al. [49]

shifted to performing refinements on complete continuous latent variables, yielding a better trade-off between quality and speed.

A variant of INAD is semi-autoregressive decoding (SAD). SAD combines autoregressive and non-autoregressive generation jointly to train the fast-decoding model. Wang Chunqi et al. [50] first proposed semi-autoregressive neural machine translation by generating phrases autoregressively in global while maintaining non-autoregressive properties inside the phrase. During inference, the decoder generates several consecutive words within the same group parallelly and then predicts the next group conditioned on the former. Stern Mitchell et al. [51] came up with the Insertion Transformer, which permits flexible sequence generation via various inserting operations conditioned on previously initialed sequences. For example, one insertion each time step in an autoregressive manner or multiple insertions for different locations in a non-autoregressive manner. Furthermore, Kasai Jungo et al. [52] put forward the Disentangled Context (DisCo) transformer, allowing simultaneous prediction of current tokens conditioned on arbitrary subsets of other tokens rather than preceding context generated before. Differing from Wang Chunqi et al. [50], RecoverSAT, proposed by Ran Qiu et al. [53], separates the target sequence into a few segments and performs non-autoregressive prediction at the segment level. The model autoregressively produces tokens within a single segment based not only on other tokens inside the same group but also on tokens in other segments. Guo Pei et al. [54] adopted two decoding modules: the former generates coarse translation that is then renewed by the latter, both of which are performed in one pass.

Another variant of INAD is conditional masked non-autoregressive decoding (CM-NAD). The masked language model stems from the pre-training strategy of BERT [10], which pretrains the model to recover the portion of words that have been randomly masked in source input by jointly considering the observed context. Ghazvininejad Marjan et al. [55] introduced a conditional masked language model (CMLM) for the first time to non-autoregressive neural machine translation by repeatedly masking out and generating words in parallel that the model has the least confidence in. The mask-predict framework first produces fully masked target tokens parallelly as a coarse version when decoding, and then it masks out words in which the model is least confident in and predicts them based on the unmasked subset of words in current translation. This mask-predict cycle keeps executing circularly until termination is met. Inspired by CMLM, Kreutzer Julia et al. [56] explored better inference strategies for CMLM by forbidding re-masking words that have been masked before and selecting positions to be masked out via certain probability thresholds. Xiao Yisheng et al. [57] further introduced an adaptive masking strategy to strengthen the refinement capability of CMLM's decoder.

3.2. Training Arrangement

As the aforementioned analogy indicated, dependency management provides a distinct way to require these experts in the translation panel, considering the learning space of target side sequences. This stands for the perspective of how to better make use of source inputs to accomplish the common translation task. However, in consideration of the interior constituents of the panel itself, there can be a discrepancy between different experts. To be specific, experts with superior translation capacities may submit a better answer and contribute positively to this mission. On the contrary, the inferior one with limited capabilities is likely to impose negative degeneration on the whole translation by offering repeated, missing, or wrong tokens. Despite this, and different from dependency management, another feasible therapy for NAMT is to implement more effective training for experts inside the panel to promote their competence and narrow the capacity gap between them. This branch of work focuses on various training arrangements, including new training objectives and multiple training strategies.

3.2.1. New Training Objectives (NTO)

Both AMT and premier NAMT use word-level cross-entropy (XE) as a loss function guiding the model's training. At each token's position, in order to maximize the likelihood probability, XE draws on one-to-one accurate alignments between the outputs and standard references. This can be seamlessly tractable for AT, as previously generated tokens provide a precise scaffold to model the offsets. However, it is almost impossible for NAT to accurately capture the token matching due to the lack of target sequential dependencies. Some researchers attributed the quality gap between AT and NAT to the improper loss function used and thus proposed new training objectives (NTO) to guide the training of NAT.

Libovick Jindrich et al. [58] first replaced the XE with connectionist temporal classification (CTC) [59] as a loss function to guide the training of the NAT model. Utilizing dynamic programming, CTC is able to marginalize all possibilities for output sentences without rigid one-to-one alignments, akin to XE, which accommodates more viable schemas. Wang Yiren et al. [60] devised two delicate regularization terms, i.e., similarity regularization and reconstruction regularization, to tackle repeated translation caused by indistinguishable adjacent hidden states in encoder outputs and incomplete translations incurred by incomplete semantic information representing hidden states in decoder inputs. Li Zhuohan et al. [61] designed two kinds of hints to encourage the NAT to imitate the consecutive hidden states and attention distribution inside the decoder layers of the AT teacher. Shao Chenze et al. [62] proposed to construct target-side dependencies by minimizing the bag-of-n-grams (BoN) differences between the inference outputs and reference sentences instead of XE, allowing phase-level approximations rather than restricting word-level alignments. Ghazvininejad Marjan et al. [63] came up with aligned cross entropy (AXE), a new loss function alternative to XE that assigns loss based on monotonic word matching among target outputs and inference sentences. This encourages the NAT model to center on radical errors such as words missing or mistaking rather than penalizing heavily on local token position caused by slight word order shifts, which exerts almost-little effects on semantic meanings. Tu Lifu et al. [64] deployed a pretrained autoregressive teacher model to define the energy, and then the non-autoregressive student is regarded as an inference network to be trained to minimize the energy of the AT teacher. Du Cunxiao et al. [65] facilitated this thread further through Order-Agnostic Cross Entropy (OAXE) to remove penalties on word order errors that were fundamentally adopted in XE. The strategy results in a more relieved loss by distributing the loss in terms of best alignments, as well as lexical matching, between target predictions and reference sequences. Shao Chenze et al. [66] integrated and extended two preceding conference works [62,67] by incorporating sequence-level training techniques and a novel loss function through the bag-of-n-grams method to evaluate achieving a compositive decoding speed and quality balance.

3.2.2. Multiple Training Strategies (MTS)

Apart from exploiting new objectives, some work turns to transferring training strategies proven to be effective in other domains to NAMT, including conventional machine learning techniques and emerging pretrain–finetuning language learning methods.

As for conventional machine learning techniques, Wei Bingzhen et al. [68] adapted the intuition of imitation learning to a non-autoregressive scenario. By using an AT demonstrator to supervise the states of a NAT learner at each decoding step across all decoding layers, the NAT learner is forced to imitate the intermediate representations of AT decoders. Sun Zhiqing et al. [69] incorporated a linear chain of conditional random fields (CRF) to model richer distributions on the target side by formulating the sequence generation problem as a sequence labeling task. Inspired by the region of automatic speech recognition (ASR), Zdenek Kasner and Jindrich Libovick'y et al. [70] combined CTC with beam search in an n-gram language model to improve the coherence of target-side generations of NAT. Guo Longteng et al. [71] applied the reinforcement training paradigm, formulated non-autoregressive generation as multi-agent reinforcement learning, and came up with Counterfactual-Critical Multi-Agent Learning. Shan Yong et al. [72] made use of model

coverage information that has been effectively applied in autoregressive generation to discriminate the untranslated subsets of words from source inputs and demonstrate the model if the source token is translated or not to improve the coherence of non-autoregressive generations.

About pretraining and fine-tuning methods, widely used in the computer vision domain, Guo Junliang et al. [73] united this paradigm with a curriculum learning method to progressively switch the knowledge and generalization ability of an AT model to a NAT model. Furthermore, Liu Jinglin et al. [74] utilized task-level curriculum learning to shift from AT generation, an easier task, to NAT generation, a harder task, using an intermediate task with an adaptive hyperparameter K . Apart from indicating the extent of parallelism, the K is used to manage and smooth the transfer process. Guo Junliang et al. [75] directly incorporated the strong pre-trained language model BERT to realize non-autoregressive sequence generation. Through two delicately devised adapters, the pre-trained BERT is adapted to the NAMT task as an encoder and decoder, respectively. Su Yixuan et al. [76] continued this incorporation between the BERT model and non-autoregressive generation, and an extra CRF layer was additionally appended to better capture target-side dependencies. Li Pengfei et al. [77] proposed CeMAT, a multilingual conditional masked pre-trained language model in which the encoder and decoder are trained with MLM and CMLM, respectively. To initialize the encoder and decoder of NAT and fine-tune them on the corresponding datasets, the model yields considerable performance progress.

3.3. Multi-Mechanism Integrated (MI)

In addition to the methods above, there are also some combinations with multiple mechanisms and methods to improve the performance of NAMT.

For FNAD, Guo Junliang et al. [78] proposed to enhance the decoder inputs in two aspects. One was to use a pre-trained phrase table to generate a coarse translation as decoder input. Another was to map the source embedding to the target embedding by optimizing the L2 distance at the sentence level and the adversarial loss at the token level. Gu Jiatao et al. [79] explored the combinations of multiple methods to relax the dependency on the target side in terms of four dimensions, including the training corpus, model architecture, training objective, and learning strategy. Huang Fei et al. [80] proposed a DA-Transformer and utilized a directed acyclic decoder to capture the translation multimodality, yielding competent performance. Shao Chenze et al. [81] then used a Viterbi decoding framework to promote the decoding accuracy of the DA-Transformer. Ma Zhengrui et al. [82] turned to modeling a fuzzy alignment between the directed acyclic graph paths and reference translations. Huang Fei et al. [83] attempted to pre-train DA-Transformer to adapt to a wider range of text generation tasks beyond machine translation. Shao Chenze et al. [84] introduced a specialized module to rephrase the reference translations to fit the NAT output for model training, realizing a better balance between speed and quality.

Additionally, concerning INAD, several proposed ideas were integrated together to recapture the lost target sequential information. Inspired by the masked language model and curriculum learning framework, Qian Lihua et al. [85] utilized both in the proposed Glancing Language Model with Glancing Transformer to shift the procedure of iterative refinements from the inference stage to the training phase. By replacing the masked token with the corresponding context source embedding, all unsampled tokens are predicted in one pass. Xie Pan et al. [86] added two consistency regularization terms to the CMLM model and yielded slight improvements. Savinov, Nikolay et al. [87] unfolded the process of denoising the autoencoder with the Markov Chain to refine the prediction in an unrolled step-by-step manner conditioned on the corrupted portions of the sequence. Huang Chenyang et al. [88] switched to construct a variant of the iterative decoding paradigm from the angle of the model's architecture. The decoder retains autoregressive decoding in the former layers while keeping non-autoregressive decoding in the last layer by generating words in parallel. Though both AT and NAT models are jointly trained to collaborate, Wang Minghan et al. [89] aim to perform predictions based on different contexts, and Ge,

Tao et al. [90] adopted a draft-verification pattern. In addition, Qin, Bo et al. [91] conducted a probe into a better trade-off between speed and quality when there were batches of sentences processed. Wang Xinyou et al. [92] introduced accessorially weaker AT decoders to strengthen the NAT models accumulatively, yielding decent performance progress. All of those works tried to complement the superior generation quality of the AT with the preponderance of the latency of the NAT.

Towards NTO, Saharia Chitwan et al. [93] attempted a CTC loss function with Imputer Transformer [94], used in automatic speech recognition, to model monotonically latent alignments between prediction and reference by marginalizing over all possible alignments. Zhang Kexun et al. [95] combined the CTC with OAXE and came up with a new objective to address the syntactic multi-modality of the translations. In addition, Shao Chenze et al. [96] explored non-monotonic alignments based on CTC loss to achieve competent performance. Du Cunxiao et al. [97] extended OAXE to phrase-based OAXE by allowing for reordering among n-gram phrases. Li Yafu et al. [98] proposed multi-granularity optimization for NAT, accommodating various dependency evaluations and obtaining competitive results.

4. Quantitative and Qualitative Analysis

In this section, we first briefly introduce the model architectures of various methods elaborated on in Section 3 and present their performances according to the reported BLEU [13] scores in the form of quantitative graphs. Finally, a qualitative analysis will also be provided.

4.1. Model Architectures and Results Overview

4.1.1. Transformer

Transformer adopts a self-attention-based encoder–decoder framework. The encoder part consists of six identical encoders, and each contains two sub-layers, i.e., a feed-forward neural network and a self-attention layer, with a residual connection and a layer normalization operation inserted between them. The decoder part keeps the same structure, except that an extra masked encoder–decoder attention layer is attached to each decoder before the two sub-layers. Figure 3 depicts the transformer architecture.

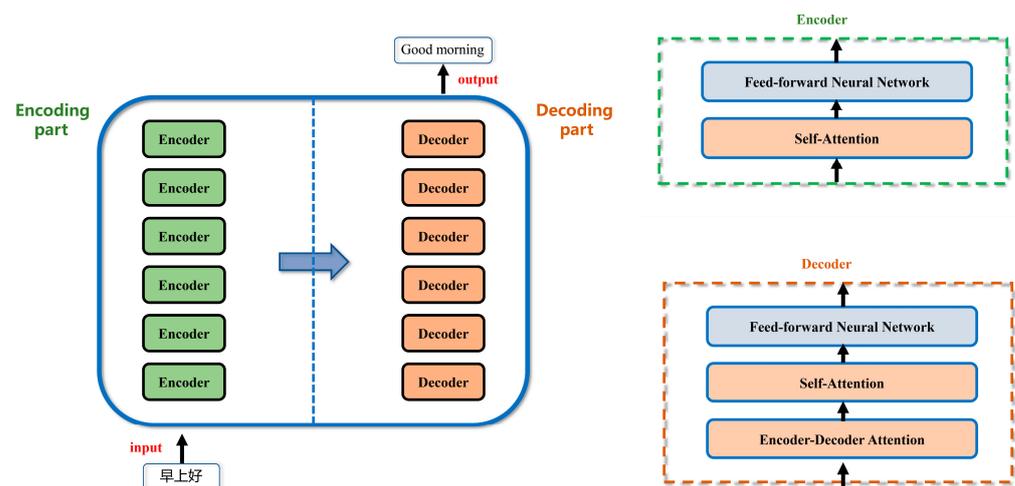


Figure 3. The architecture for Transformer, its encoder and decoder, respectively. To simplify, residual connection and layer normalization operations have been omitted.

During inference, the source inputs are first encoded into hidden representation vectors in high dimensions by the encoder, and then the decoder predicts each target word autoregressively conditioned on the source inputs and the hidden states.

The key to the splendid accuracy of the Transformer model is the attention mechanism. A Linear transformation is performed on the matrix composed of input vectors obtained through source embedding to produce three matrices, Q , K , and V , parallelly. After the dot

product operation, the Q and K matrices are scaled according to the predefined dimension. The results are then normalized through a softmax function to obtain the attention weight matrix, which is multiplied by matrix V to produce the final outputs. This process can be formulated as the equation below:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where $\sqrt{d_k}$ is the dimensionality of matrix Q and K , and softmax means the softmax normalization function.

4.1.2. Latent-Variable-Based Model for FNAD

For FNAD, the latent-variable-based model prevails due to its distinct transition flow from source side to target side. The source inputs are first encoded into source embedding representations. Based on that, corresponding latent variables are produced. The decoder finally predicts all the target tokens in parallel, conditioned on the latent variables and source inputs. Due to the lack of previously generated tokens as a posterior condition, it is hard for the NAT model to consider the multimodal distribution of target sentences based merely on input information. With the use of latent variables, feasible posterior conditions are supplemented to provide richer supervision, bridge the tough mapping gap between source and target, and allow tractable searching in output space.

As Figure 4 shows, the encoder of this model category stays unchanged as Transformer, whereas the casual restriction in the masked encoder–decoder attention layer is abandoned to enable future attendance across all positions. Positional encoding is also needed to capture reorder information. Other than that, an additional module or specialized technique is required in order to generate proper latent variables. Specifically, three types of latent variables are most commonly used: discrete latent variables constructed by a particular predictor using a discretization technique such as VQ-VAE with variational inference; syntactic and semantic prior knowledge generated through an external parser or appended neural networks; and reordering or position information modeled by a predefined predicting module.

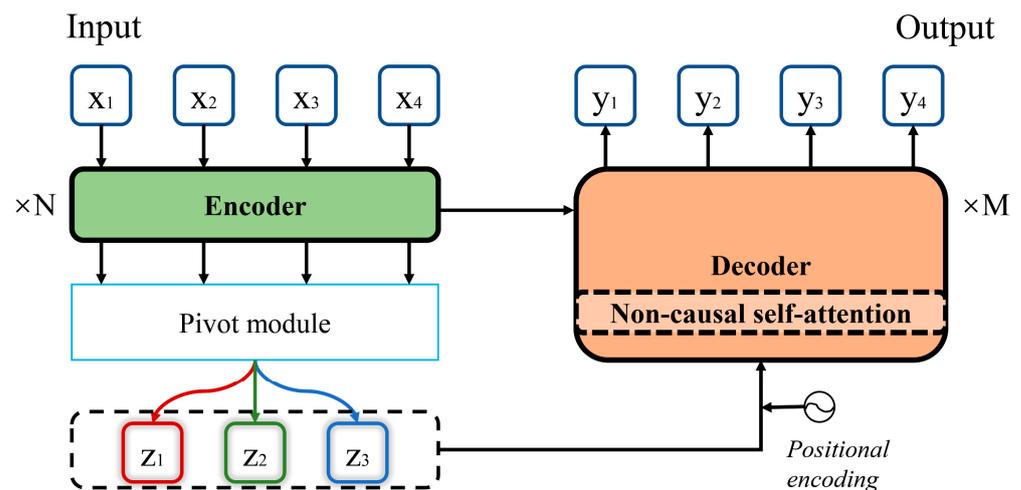


Figure 4. The architecture example for FNAD. N and M means stacks.

4.1.3. Iterative Model for INAD

INAD sacrifices a certain speed to balance the trade-off between translation accuracy and decoding latency. The speed loss mainly derives from the iterative refinement process induced by multi-step decoding. Using the former generated sentences as posterior knowledge to accumulatively approach the golden distribution, the current decoding pass

produces outputs conditioned on source inputs as well as outputs from the previous pass when decoding. The process can be regarded as an autoregressive generation paradigm at the sentence level. This thread of the model can be divided into single-decoder iterations and multi-decoder iterations, according to whether the refinements of multiple steps are performed within the same decoder or not. As described in Section 3.1.2, the original iterative refinement decoding by Lee Jason et al. [47] adopted two decoders to achieve multi-step decoding.

In this framework in Figure 5a, decoder 1 produced a coarse version of the translation used for the following refinements, which happened in the second decoder. The final generation came from multiple iterations of coordination via both decoders. Most other refinement policies, such as SAD and CMLM, generally executed multiple iterations via one single decoder with the necessary modifications to the vanilla Transformer model, including removing the self-attention mask and appending positional embedding. Despite the similarity of the model’s architecture, as Figure 5b depicts, the difference mainly stems from decoding policies. SAD combines local autoregressive generation with global non-autoregressive generation or vice versa, whereas CMLM utilizes a mask-predict paradigm to realize parallel predicting.

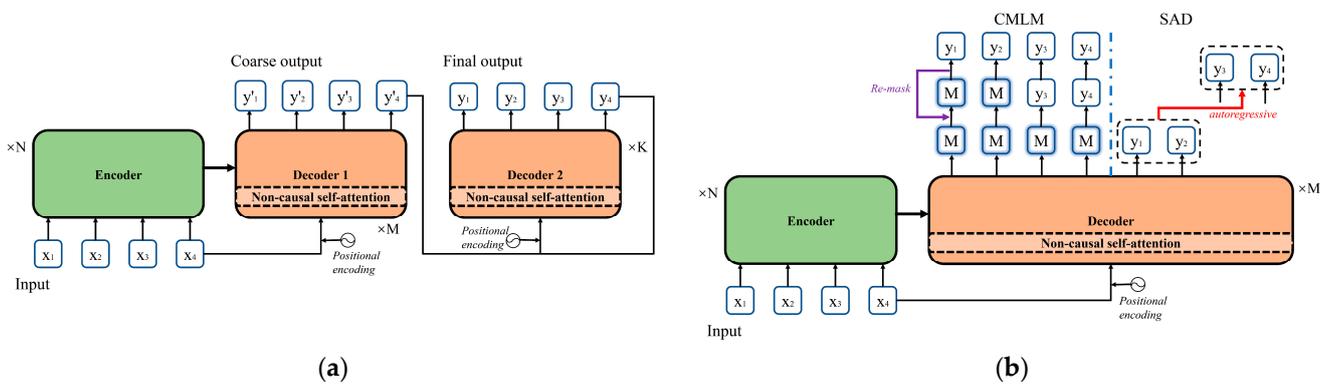


Figure 5. The architecture example for INAD: (a) original iterative refinement model, (b) conditional masked language model (CMLM), and semi-autoregressive model (SAD).

4.1.4. Models with New Training Objectives (NTO)

Training with cross-entropy yields significant accuracy in AT. The golden prediction can be made in order to minimize the training loss between ground truth and model predictions because previously generated words offer certain posterior restrictions. However, the same objectives degenerate in NAT, as it is intractable for the model to tackle the exponentially increased output space of non-autoregressive generation. In addition, reordering information is amplified by the unrestricted parallel decoding of NAT, which does not happen in AT. This results in potential error alignments and weak correlations between the loss function and the translation quality. To alleviate this inconsistency, on the one hand, some new objectives guide the model to focus on the whole semantic coherence at the sentence level, such as CTC [59] and BoN [62], thus bypassing restricting word-to-word alignment. On the other hand, some models are encouraged to imitate the hidden representation from AT teachers such as hint-based and auxiliary regularization, therefore avoiding modeling the complicated reordering information of output space. Figure 6 shows an instance of a model with new training objectives.

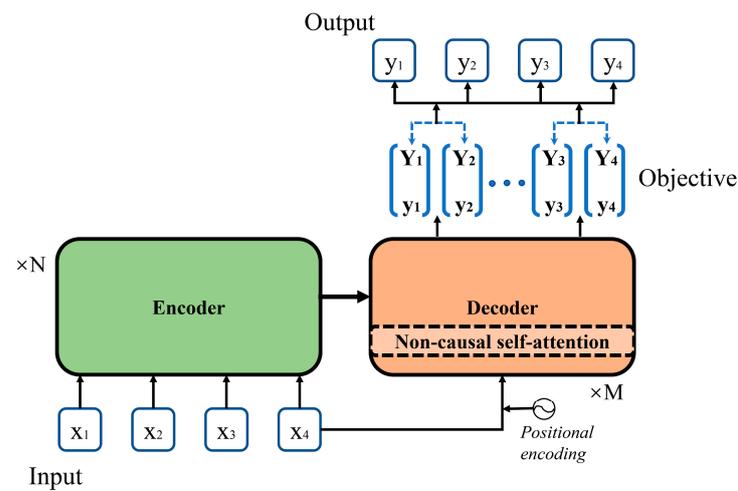


Figure 6. The architecture example for NTO.

4.1.5. Models by Multiple Training Strategies (MTS)

To simplify, this series of models trained by multiple training strategies can be roughly divided into two categories: One of them is a model incorporating conventional machine learning techniques such as CTC and CRF [99]. Another utilizes pre-trained and fine-tuned exemplifications such as BERT [10]. The former adopts a similar encoder structure akin to the AT model but implements some advisable modifications to adapt the decoder for the appended CRF module. For the latter, to apply the pre-trained and fine-tuned techniques to the NAT models, specialized neural networks such as encoder–decoder adapters and delicate transferring methods such as curriculum learning policies are implemented. See an approximate demo in Figure 7 for this line of model.

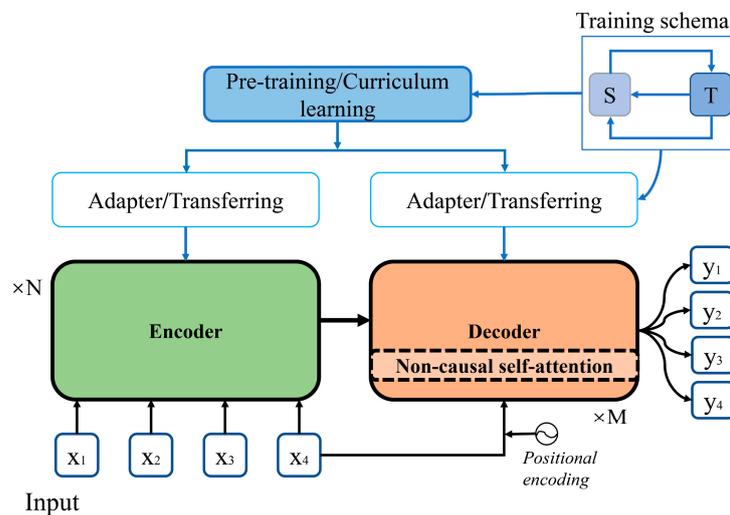


Figure 7. The architecture example for MTS.

4.1.6. Results Overview for All Models

We selected the best performance from the reported work for each model category to offer a results overview for all models in Table 2. The autoregressive baseline, basal performance, and optimum performance are listed in the left, middle, and right order for each column. BLEU gains are obtained by subtracting the autoregressive baseline from basal and optimum performance separately. Table 2 presents a global glimpse of the best results for all model categories, and a more specific and comprehensive performance description will be detailed in Section 4.2.

Table 2. An overview of the best results achieved by each of the mentioned models.

	WMT14 En-De	Latency Speedup	BLEU Gains
FNAD-[43]	27.48/24.64/27.50	1.00×/22.6×/9.3×	−2.84/+0.02
INAD-[53]	27.17/26.32/27.11	1.00×/4.31×/2.16×	−0.85/−0.06
NTO-[66]	27.42/23.90/25.54	1.00×/15.6×/15.6×	−3.52/−1.88
MTS-[75]	28.04/28.08/28.69	1.00×/4.7×/2.4×	+0.04/+0.65
MI-[90]	27.38/28.73/28.89	1.00×/3.0×/3.6×	+1.35/+1.51

4.2. Performance Description

This segment will provide a quantitative description of the various model endeavors toward NAMT mentioned above. In order to assess the performances of the proposed works, we quote four strong candidates, as well as the baseline work for each model category in terms of the reported BLEU score.

The original NAT model is trained and validated on three types of languages, covering English, German, and Romanian. The BLEU scores are evaluated on official test sets, i.e., newstest2014 for WMT English-German (<http://www.statmt.org/wmt14/translation-task> (accessed on 20 June 2023)) and newstest2016 for WMT English-Romanian (<http://www.statmt.org/wmt16/translation-task> (accessed on 20 June 2023)) or the development set for IWSLT16 English-German (<https://wit3.fbk.eu/> (accessed on 20 June 2023)). Though BLEU scores vary moderately with regard to different languages (an average difference of 5.9) and even different translation directions (an average difference of 2.4), the tendencies of translation accuracies and speedup gains are considerably consistent across all languages and datasets. As the task mainly investigates the implications of non-autoregressive generation patterns on the sphere towards decoding speedup and quality changes, the selection of datasets and language types exert limited impacts on the performance evaluations for the proposed approaches. In addition, succedent works may validate their models on various datasets, but the newstest2014 for WMT English-German (WMT14 En-De) is most commonly used.

Therefore, without loss of generality and for the sake of simplicity and comparability, we mainly center on the most recognized dataset of WMT14 En-De to demonstrate the performance case. The reported BLEU score for each proposed paper is collected and listed according to the left, middle, and right order, which manifests the autoregressive baseline, basal performance with fundamental setting up, and best performance with optimal deployment, respectively. In addition, the corresponding latency speedup is also exhibited separately with respect to the model’s setup. To better compare the accuracy of different models, the BLEU score gains are additionally listed according to their clean value. The value is obtained by subtracting the AT baseline from the BLEU score of basal and optimum setting up. Tables 3–7 refer to the fully non-autoregressive model (FNAD), iteratively non-autoregressive model (INAD), new training objectives (NTO), multiple training strategies (MTS), and multi-mechanism integrated (MI), respectively.

Table 3. The translation accuracy and latency for FNAD. The autoregressive baseline, basal performance, and optimum performance are listed in the left, middle, and right order for each column. BLEU gains are obtained by subtracting autoregressive baseline with basal and optimum performance separately. AVG means the average value. “null” means the data are not reported.

	WMT14 En-De	Latency Speedup	BLEU Gains
FNAD-base [12]	23.45/17.35/19.17	1.00×/15.6×/2.36×	−6.10/−4.28
[43]	27.48/24.64/27.50	1.00×/22.6×/9.3×	−2.84/+0.02
[44]	27.33/25.56/26.60	1.00×/10.37×/5.59×	−1.77/−0.73
[40]	27.40/25.70/26.40	1.00×/13.9×/13.0×	−1.70/−1.00
[88]	27.48/ null /27.02	1.00×/ null /14.8×	null/−0.46
AVG B	-	-	−3.10/−1.29
AVG S	-	1.00×/15.62×/9.01×	-

Table 4. The translation accuracy and latency for INAD.

	WMT14 En-De	Latency Speedup	BLEU Gains
INAD-base [47]	24.57/13.91/21.61	1.00×/8.9×/1.2×	−10.66/−2.96
[50]	27.11/23.93/26.90	1.00×/5.58×/1.51×	−3.18/−0.21
[53]	27.17/26.32/27.11	1.00×/4.31×/2.16×	−0.85/−0.06
[49]	28.30/25.70/27.40	1.00×/15.0×/6.20×	−2.60/−0.90
[57]	28.41/ null /27.57	1.00×/ null /2.3×	null/−0.84
AVG B	-	-	−4.32/−1.00
AVG S	-	1.00×/8.45×/2.67×	-

Table 5. The translation accuracy and latency for NTO.

	WMT14 En-De	Latency Speedup	BLEU Gains
NTO-base [58]	22.94/12.51/17.68	1.00×/5.8×/3.4×	−10.43/−5.26
[60]	27.30/20.65/24.61	1.00×/27.6×/15.1×	−6.65/−2.69
[61]	27.30/21.11/25.20	1.00×/30.2×/17.8×	−6.19/−2.10
[62]	24.57/16.05/20.90	1.00×/10.76×/10.77×	−8.52/−3.67
[66]	27.42/23.90/25.54	1.00×/15.6×/15.6×	−3.52/−1.88
AVG B	-	-	−7.06/−3.12
AVG S	-	1.00×/20.00×/12.53×	-

Table 6. The translation accuracy and latency for MTS.

	WMT14 En-De	Latency Speedup	BLEU Gains
MTS-base [68]	27.41/22.44/24.15	1.00×/18.6×/9.70×	−4.97/−3.26
[69]	27.41/20.27/26.80	1.00×/14.9×/4.39×	−7.14/−0.61
[73]	27.30/21.70/25.75	1.00×/28.9×/16.0×	−5.60/−1.55
[74]	27.30/21.94/25.37	1.00×/27.6×/16.0×	−5.36/−1.93
[75]	28.04/28.08/28.69	1.00×/4.7×/2.4×	+0.04/+0.65
AVG B	-	-	−4.61/−1.34
AVG S	-	1.00×/18.94×/9.70×	-

Table 7. The translation accuracy and latency for MI.

	WMT14 En-De	Latency speedup	BLEU gains
MI-base [78]	27.41/20.26/24.28	1.00×/24.3×/12.4×	−7.15/−3.13
[93]	27.80/25.80/28.20	1.00×/18.6×/3.9×	−2.00/+0.40
[79]	27.48/19.50/27.49	1.00×/17.6×/16.5×	−7.98/+0.01
[87]	27.30/27.94/28.46	1.00×/4.7×/1.4×	+0.64/+1.16
[90]	27.38/28.73/28.89	1.00×/3.0×/3.6×	+1.35/+1.51
AVG B	-	-	−3.03/−0.01
AVG S	-	1.00×/13.64×/7.56×	-

In terms of translation quality, we can see from all tables that the basal performances all degenerate to the optimal setting up, as the latter demands more complicated technique deployments and calculating costs. Due to the different model deployment of AT baseline and Graphical Processing Unit (GPU) platforms, the reported BLEU scores could be various, and it is improper to directly compare different works merely based on the cited BLEU value. We consider the BLEU gains a better criterion in spite of their own AT baseline. As shown in Table 7, four of the five cited MI models all outperform the AT baseline, most by 1.51 points. MI achieves the best translation performance with the highest average BLEU gains of −0.01, compared to other model categories. The second place is the INAD models of Table 4 with the average BLEU gains of −1.00. By contrast, the NTO models demonstrate the worst translation quality according to the lowest average BLEU gains of −3.12. The other two, FNAD and MTS, are in third and fourth place with similar average values of −1.29 and −1.34. In addition to the NTO model, the rest of the four categories are able to

catch up with or even surpass their AT baseline. This phenomenon supports the conclusion that the NTO model is globally inferior to other models in terms of translation quality. As for latency speedup, the five categories all achieve significant decoding speed gains beyond their AT baselines in all tables. However, the speedups of models under optimal setups almost lag behind those with basal deployment. This may empirically be caused by the fact that the former arrangement consumes extra modules to achieve better accuracy, generally leading to additional latency and slowing down the decoding process. In general, the NTO models realize the fastest speedup times beyond all the other categories, with largest average latency speedups of 20.00 and 12.53, according to Table 5. The MTS achieves the second acceleration with average speedups of 18.94 and 9.70, which were followed by the FNAD with lower speedups of 15.62 and 9.01, separately, in Tables 3 and 6. The subsequent model is MI, with inferior speed values of 13.64 and 7.56 compared to the former two. Not surprisingly, the INAD models in Table 4 fell behind previous categories by a large margin, with the lowest speedup gains of 7.76 and 2.8. These were consistent with the iterative property of such a method. Additionally, several works, such as [62,66,79,90], still maintain the latency superiority of basal setting up and approach superior accuracy even when optimal deployment is arranged.

4.3. Performance Analysis

On top of the performance description above, a more in-depth analysis of each model category is elaborated on in this subsection. To this end, we first score each model category according to their rankings of translation quality and latency speedup for a comprehensive evaluation across all types. Next, to shed more light on the intrinsic properties of all categories, the superiority and deficiency of each are also detailed, as are further discussions.

Firstly, we assigned a performance score (P score) for each model category to denote their compositive performance capabilities in terms of their rankings of translation quality and latency speedup. As Table 8 shows, the assigned scores were inversely proportional to their rankings. For instance, the top category obtained 5 points, whereas the last one obtained 1 point, and so on. Additionally, the final P score was summed of their own ranking score from translation quality and latency speedup. Then, the ultimate P score for each model type was $P(\text{MI}) = 5 + 2 = 7$, $P(\text{FNAD}) = 3 + 3 = 6$, $P(\text{NTO}) = 1 + 5 = 6$, $P(\text{MTS}) = 2 + 4 = 6$, and $P(\text{INAD}) = 4 + 1 = 5$. The histogram in Figure 8 depicts a global performance evaluation across all categories.

Table 8. The performance score for all model categories.

Ranking	Translation Quality	Latency Speedup	Score
1	MI	NTO	5
2	INAD	MTS	4
3	FNAD	FNAD	3
4	MTS	MI	2
5	NTO	INAD	1

In general, the MI earns the highest P score, i.e., 7, and shows the best non-autoregressive translation performance. Although it is not precisely in line with the authentic situation, the fact that the FNAD, NTO, and MTS obtain identical P scores manifests a similar translation capacity tendency for the three model types. Moreover, the FNAD achieves an optimal quality–speed balance through the same P1 and P2 scores, which means its translation accuracy is well associated with speedup. The largest difference value (namely, 4) between P1 and P2 stems from NTO, revealing an uncoordinated correlation between translation quality and latency speed. Moreover, with the lowest P score of 5, INAD exhibits inferior translation ability compared to others, which suggests that the inspiration of trading speed for accuracy yields no sparkling outcome but limited benefit.

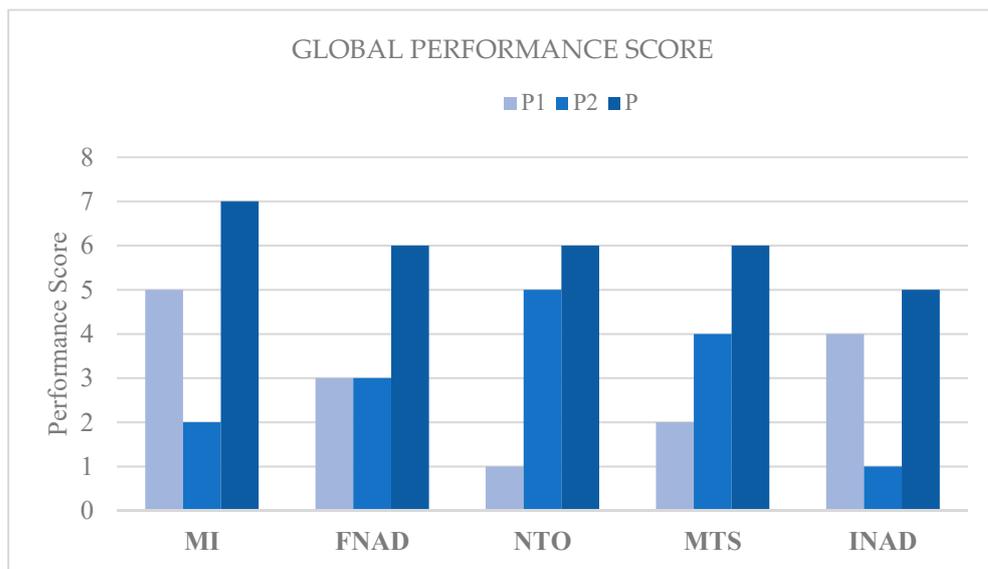


Figure 8. The histogram of global performance evaluation for all model categories. P1 and P2 denote the score of translation quality and latency speedup separately. P is the composite performance score, and $P = P1 + P2$. MI, FNAD, NTO, MTS, and INAD are the abbreviations of each model category.

To accommodate more insightful views, we inspect all categories from four aspects: what are the basic properties? What is it able to do? What is the advantage? What is the flaw? For the purpose of qualitative comparison, Table 9 provides a composite summary of those four dimensionalities.

Table 9. A summary and qualitative comparison for all model categories.

Methods	Properties	What can do	Advantage	Disadvantage
MI	Multi-mechanism integrated	Allow multi-mechanism learning	Complement each other’s superiority	Need to integrate multi modules
FNAD	Pivot and latent-variable-based	Model any types of dependencies	Concise transfer flow	Model is hard to train and converge
NTO	Multimodality alignments regularized	accommodate coarse-grained match	Model training is end-to-end without intermediates	Lack of strong performance modules
MTS	Cross-domains transfer of ready methods	Enable one-stop training	With established performance basics	Require delicate adapting schema
INAD	Multi-pass decoding	Sentence-level autoregressive generation	Mechanism is distinct and tractable	Repeated decoding lags speed

As an example, from the perspective of the task itself, FNAD represents the original enlightenment of NAMT and is characterized by totally non-autoregressive generation via one-pass decoding. By using latent variables and other alternative pivots, it is plausible for FNAD to model any dependencies and bridge the omitted posterior gap between NAT and AT. However, generating the proper discrete latent variables demands a complicated discrete bottleneck that is indifferentiable and cannot be optimized by the effective gradient propagation algorithm. Though this can be addressed by Evidence-Lower-Bound optimization, the model will be hard to train and converge.

Scarifying speed with quality can be regarded as another natural idea, but this trade-off schema is likely to be limited by the dual contradiction. The appended decoding step can successively promote the quality of the original translation, whereas generation

latency is also induced. It is hard to tell which benefits INAD more. By contrast, with the use of the new loss function, NTO enables one-stop model training without relying on any intermediate modules in an end-to-end manner, which bypasses the knotty balance problem. MTS resorts to relevant tasks and transfers the methods effectively utilized in other domains where delicate adapters or operational training schemas need to be devised. Other than that, it seems to be a promising way to integrate multiple effective mechanisms and enable MI to learn from other methods. To sum up, we list the following conclusions:

- (1) On behalf of the original NAMT, FNAD achieves a favorable balance between translation quality and speed, though it is hard to train the model.
- (2) Restricted by the compromising contradiction, INAD yields limited benefits by substituting speed for quality.
- (3) NTO allows one-stop model training and bypasses the trade-off problem, but strong modules or means are needed to further promote translation accuracy.
- (4) MTS and MI all learn from other effective methods. However, the former focuses on correlative domains and tasks, thus requiring an extra adapting module or training schema. Additionally, the latter mainly integrates available mechanisms from the same field to complement each other. In general, MI seems to be a more promising way to gain further progress by combining the superiorities of others.

5. Problem Inspection

Even though a comprehensive analysis of all model categories is illustrated in Section 4, we mainly center on separate properties for each, putting aside potential difficulties and challenges they may encounter in common. This section will inspect two general problems facing all models: (1) target sentence length prediction and (2) sequence-level knowledge distillation and explore probable therapies for them.

5.1. Target Sentence Length Prediction

Due to the autoregressive operation during inference, the AT model can determine the length of the target sequence adaptively. However, this adaptive mechanism does not exist in NAT because the dependencies among target words are discarded, and there is no iterative pattern in non-autoregressive decoding. Figure 9 illustrates the inference framework of the target sentence for both.

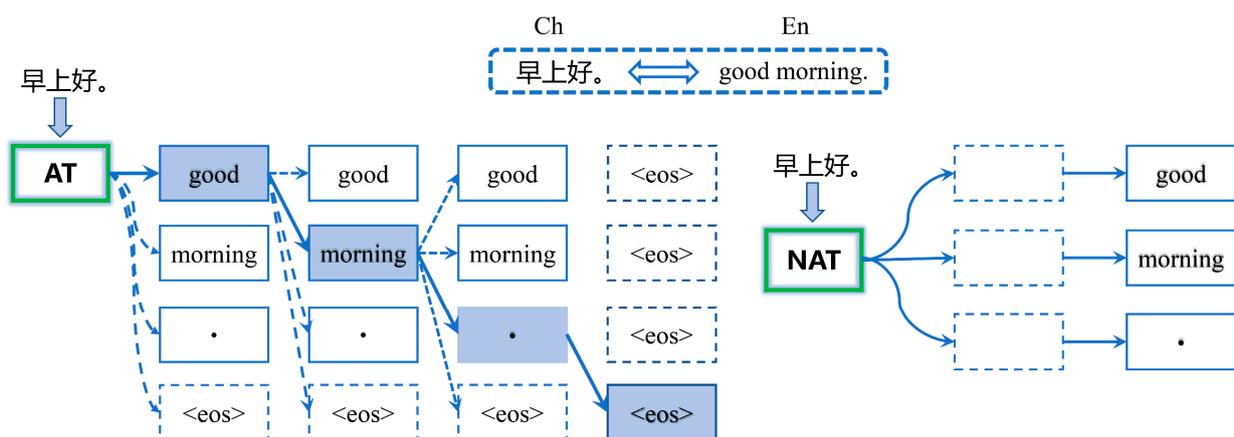


Figure 9. An example about the process to determine target length of AT and NAT.

Specifically, during the running period, the AT model takes into account the recently generated word and predicts the next. For each step, the word with the highest probability is selected by beam search to maximize the likelihood of the whole sentence. This course dynamically ends when a stopping token is encountered. Given a source Chinese sentence “早上好” and its target translation “good morning”, after the word “morning” is generated,

the <EOS> token with the highest probability will be selected to output, which stands for the termination symbol of the sentence.

By contrast, as all words are produced in parallel, there is no suitable stop threshold in the NAT. Therefore, the length of the target sentence needs to be determined in advance. As the picture shows, before decoding “good morning”, the most probable target length is predicted as three, which functions as a scaffold and accommodates for the subsequent generation for the entire sequence. For instance, in the original literature of NAMT, Gu et al. [12] utilized the fertility to copy source words as decoder inputs while tactfully determining the target lengths. By specifying a fertility value for every word in the source input, the final length of the target output can be ascertained by calculating the sum of all fertility values for the whole source sequence. The prediction of fertility sequence towards the source sentence is accomplished by an exclusive module, which is composed of a one-layer network with a softmax classifier and appended on top of the last encoder layer.

Following this practice, the common approaches used in subsequent literature to acquire the target sentence length mainly fall into two categories. One is to ascertain the exact length via an exclusively trained length predictor [12,46,48,67–69,71]. Another is to predict the difference between source and target length using a specialized classifier [39,47,49,60–62,100]. However, due to the uncertainty of the target-side distribution, the target-length prediction is more akin to an explicit reflection of the multi-modality problem in NAMT’s output space. It is non-trivial to precisely finalize the accurate value for the target length. Therefore, the most recognized schema is to produce multiple available target sentences and select the best one based on several predicted target lengths.

Noisy parallel decoding (NPD) Inspired by Cho [101]’s noisy parallel approximate decoding (NAPD), Gu et al. [12] sampled multiple fertility sequences from the fertility space. By generating one translation for each fertility sequence, various translations (namely, eight candidates in this paper) can be acquired. These alternative translations are then re-scored by a pre-trained autoregressive model. The one with the highest score will be selected to be the target output as the optimal translation. As the scoring model is pre-trained, and all candidates can be generated and ranked in parallel, this process adds a little latency to the overall decoding speed. As there are multiple translations, the rest of the candidates in the translation set function as “noise” and need to be excluded compared to the final output, which is called noisy parallel decoding.

Length parallel decoding (LPD) Another widely adopted decoding strategy is length parallel decoding. Similar to NPD, LPD also generates multiple candidates and ranks them using an autoregressive teacher. Wei Bingzhen et al. [68] initially put forward LPD, which first determines a target length T_0 with a trained module and then predicts multiple target lengths varying from $[T_0 - C, T_0 + C]$, where C is a prepended constant according to searching precision. Based on all target lengths, the model generates various translations in parallel, and the optimal one is identified by the pre-trained autoregressive model. Beyond that, some researchers [47,60] employed a more distinct method. They predefined target length $T = T_S + \Delta T$, where T_S denotes the length of the source sequence and ΔT is the bias offset that can be set in terms of the overall statistical lengths of all training data. Others turned to expanding the search beams by predicting the difference in the sequence length between the source and target using a classifier ranging from $\{-B, B\}$, where B is the width of the window [39]. By contrast, [61,69] predicted a compositive length among $[T_S + C - B, T_S + C + B]$, where C is a constant term that can be determined by surveying the average difference between source and target sequence length. B is the halfwidth that manages the searching range. Contrasting with NPD, which emphasizes the prediction of the original target length, LPD pays more attention to a variational range of target length, which is called length parallel decoding.

Although it is beneficial to generate multiple candidates before identifying the optimum translation, this necessary preliminary target length prediction actually cuts both ways. We empirically observed that the two types of potential problems induced by the

length predicting operation are: (1) length exposure bias, where the target length is different between training and inference phases; and (2) additional overheads, where extra computational cost is caused by the exclusive predictor or classifier. First, with the existence of standard references, the model uses the ground truth length of the target side to guide the parallel generation of the NAMT in the training stage. However, during inference, this ground-truth length is unavailable. The model has to use the predicted one, leading to an inconsistency between the training and reasoning phases, which we call length exposure bias. As the target length works as a scaffold for later parallel decoding, potential problems could be caused by this inconsistency, as well as other de facto accuracy errors. Second, predicting target length with a specialized predictor or classifier becomes a general practice for most published papers. In view of the fact that different target lengths have a significant influence on model performance, it is important to predict the target length as accurately as possible. This, consequently, increases the additional time and computational overheads required to train an exclusive module or network with high accuracy.

Some researchers proposed to add a special LEGNTH token to the encoder's input [55,64,65,75]. By taking the encoder's output as a representation, predicting the target length is akin to predicting another token from a different vocabulary. The loss of the LENGTH token is added to the cross-entropy loss from the target side, jointly optimized and integrated into the entire model's training loss, thus avoiding the expense of training a separate length predicting module. Yet, the need for length prediction still remains.

Some solutions can be explored for a more elegant pattern to bypass the uncertainty of target sequence length. In specific, one way is to utilize CTC to accommodate sufficient length changes via more many-to-one alignments. The length of mapping representations can be k times over the output sequence, which enables abundant possible alignments from input to output. In addition, strong generation models such as parameterized Markov chains can determine target-length samples from the noise distribution. A combination of CTC and other stronger models to dynamically adapt target length may offer a possible way to elude this problem for accuracy promotion.

5.2. Sequence-Level Knowledge Distillation

Another key ingredient in the training recipe of non-autoregressive models is sequence-level knowledge distillation (SKD) [102]—a variant of knowledge distillation (KD) [103]—which is employed in almost all existing NAMT literature. Inspecting the different functions of larval and adult forms of insects in the natural world, Hinton et al. [103] proposed knowledge distillation to transfer knowledge from a large teacher model with high performance to a student model that was lightweight and easy to train. An original practice to apply this intuition, first utilized by Hinton in the domain of classification tasks, is to use the label predicted by the teacher to supervise the training of the student model. The practice aims to encourage students to mimic the teacher's distribution and catch up with its capacity.

Based on that, Yoon Kim et al. extended this schema to the context of NMT and came up with sequence-level knowledge distillation (SKD) [102]. In specific, the student model is trained on a new dataset created by replacing the target side with output translations from a pre-trained teacher network. Inspired by this practice, Gu et al. [12] then adopted the SKD to mitigate the multimodal distribution of NAMT, which yielded significant improvements. Generally, the SKD is widely used in NAMT with the following two deployments: firstly, the AT teacher model with the same architecture as the NAT student is trained on raw training data. Secondly, replacing the target sentence of each source sentence in the raw data with the translation results generated by the AT teacher as the new ground truth to form a new training dataset—a distilled pseudo-parallel corpus—on which the NAT model is trained. Figure 10 simulates the establishment of the dataset.

In order to show the effect of SKD, we quoted three works along with a baseline model for each model category elaborated in Section 4 to offer a quantificational profile. Table 10 depicts the reported BLEU score, including basal performance, optimal performance, and clean gains comparing the optimal with the basal one.

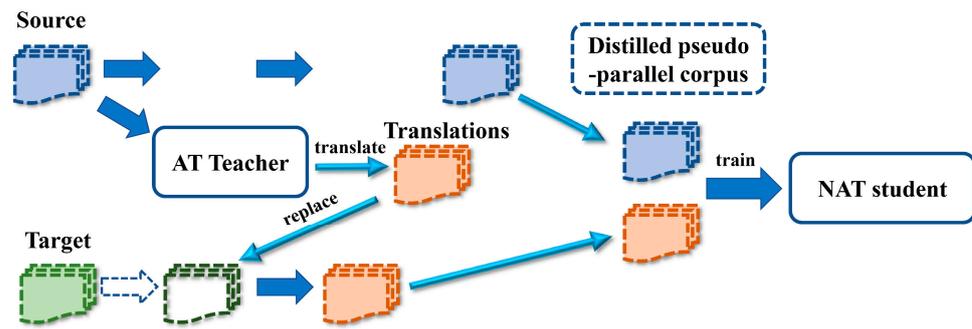


Figure 10. The establishment course of the distilled pseudo-parallel corpus.

Table 10. The reported BLEU score of some selected papers with the distillation and without the distillation. The column on the left of / represents the basal performance, and the right one denotes the optimal performance. BLEU gains are calculated by subtracting the former with the latter. The growth rate is also provided in the bracket for convenient contrast. The best scores for each item as well as the average for all BLEU gains are bold in black.

	BLEU Score without SKD	BLEU Score with SKD	BLEU Gains
FNAD			
[12]	16.51/18.87	20.72/25.20	4.21/6.33 (25%/33%)
[38]	21.40/22.40	26.40/26.70	5.00/4.30 (23%/19%)
[39]	18.55/20.85	24.15/23.72	5.60/2.87 (30%/14%)
INAD			
[47]	20.91/23.65	26.17/27.11	5.26/3.46 (25%/15%)
[51]	19.34/22.64	22.75/25.45	3.41/2.81 (18%/12%)
[55]	10.64/24.61	18.05/27.03	7.41/2.42 (70%/10%)
NTO			
[64]	8.28	14.58	6.30 (76%)
[63]	20.40	23.53	3.13 (15%)
[65]	22.70	26.20	3.50 (15%)
MTS			
[68]	16.51/23.56	20.72/28.41	4.21/4.8(25%/21%)
MI			
[93]	15.6/24.7	25.4/27.9	9.80/3.20 (63%/13%)
[79]	11.40	19.50	8.10 (71%)
[86]	22.89/24.37	26.25/27.39	3.36/3.02 (15%/12%)
Avg. gains			5.33/3.69(36%/17%)

As the table exhibits, with the use of SKD, the translation quality of the NAT can be improved by 9.8 and 6.33 points at most and by 3.13 and 2.42 points at least for basal and optimal performance, respectively. The highest increasing rate reaches 76 percent compared to the configuration without SKD. The least progress still obtains 10 percent increases in the literature. Contrasting with optimal performance, the basal one usually benefits more from improvements from SKD, probably because of the difficulty in further improving the accuracy based on the existing basal performance. The average gains for those collected papers are 5.33, 3.69, 36%, and 17%, which proves the effectiveness and significant improvements made by utilizing SKD over NAMT.

Why does NAMT benefit a lot from SKD? It is assumed that the new target side sentences generated by the AT teacher can reduce the “modes” of training data (multiple feasible translations for a source sentence), be less multimodal, and be more aligned to the source. It then helps the NAT model capture the target-side distribution more easily. This hypothesis is identified through an experiment that simulates multiple modes in the output space of NAMT. By aligning a source sentence in English to corresponding

target translations in German, Spanish, and French, respectively, the study formulates a multi-target En-De/Es/Fr corpus and explicitly includes three modes in output space. Zhou et al. [104]. Experimental visualization indicates that NAT tends to cluster more closely to a single language mode after being trained on a corpus decoded by an AT teacher yet scatters broadly across all language types when trained on the original dataset. Zhou et al. [104] also examined the impact of data complexity on model performance by generating datasets with different complexities via various AT models with diversified capacities. The study suggested that NAT models with larger parameters and higher capacities require distilled data with more complexity to achieve better translation quality. Based on the earlier discussion, Xu Weijia et al. [105] subsequently explored two types of data complexity, i.e., lexical diversity and word reordering degree. Experiments showed that decreasing lexical diversity and word reordering degrees via SKD both lowered the data complexity and helped NAT students learn better alignment between the source and target. Other than that, Ren Yi et al. [106] analyzed the attention weight ratio on the target token over that on full context when predicting a target word from the perspective of target dependencies rather than the data level. By using SKD to train the NAMT model, they observed that dependency on target tokens was reduced, encouraging the model to rely more on source sentences for target predictions.

Inspired by those analyses, Zhou Jiawei et al. [107] leveraged AT teachers to generate more target translations for a large amount of source text from monolingual corpora. The model was trained on the newly formed larger distilled pseudo-parallel corpus, which moderately improved the NAMT model's performance. Guo Jiaxin et al. [108] adopted self-distillation mix-up data to train the NAT model, yet this yielded limited progress. Shao Chenze et al. [109] came up with diverse distillation that generated multiple inference translations and selected the optimum one for model training. A similar inspiration was also adopted by Liu Min et al. [110] through selected knowledge distillation.

To sum up, it becomes an indispensable component to utilize SKD to train NAT models. This strategy, on the one hand, provides a practical way to promote translation quality and significant improvements in the model's performance. On the other hand, extra overheads for training AT models are required in order to generate new teacher translations. Running different models on various training corpora also leads to repeated operation redundancies and additional computational costs.

The training of NAT models heavily relies on the SKD strategy, and how to relieve this dependency for NAMT is still unclear. Some works [80–82] integrated directed acyclic graphs to perform inference, yielding competent performance even without the use of SKD. Training models on raw data with GAN [111] and other data augmentation techniques is perhaps a promising solution.

6. Discussions

In this section, we go through the fundamental properties of NAMT alongside the problems elaborated before and pose more critical discussions about speedup decoding. Additionally, we shed a bit of light on language settings with regard to low-resource languages, monolingual languages, and other lingual orientations towards the application of NAMT. A tentative examination of these language settings is also discussed.

Reexamined translation quality The proposal of original non-autoregressive decoding aims to facilitate fast translation by paralleling a new decoding paradigm to autoregressive generation. Subsequent studies either manage target dependency through reducing, constructing, or arranging training schemas via new objectives or multiple strategy transfers. While some attempts at mixing superiorities from these works achieve hyper-accuracy beyond the corresponding AT baseline, almost all existing literature still focuses on closing the gap between NAMT and AMT. None of them realizes a separate, thorough surpassing over the basal autoregressive Transformer, despite the fact that the translation quality of a mass of emerging models has far exceeded the latter. In addition, length prediction and sequence-level knowledge distillation boost performance for non-autoregressive models.

However, both of them demand supervision from the AT teacher model and, to some extent, already set an upper bound for the generating quality of NAMT. This prevents the models from further promoting translation accuracy.

Reexamined translation speed Despite its accuracy, non-autoregressive decoding literally accelerates the translation process, varying from 2 to 15 times faster compared to its autoregressive baselines in the original proposal by Gu et al. [12]. Generally, the latency is computed by the wall time to decode a single sentence without batches in a GPU environment. Nevertheless, a few subsequent studies empirically observed that the latency varied when measured in different hardware environments and application scenarios [112–114]. More precisely, in line with previous works on deep encoders or shallow decoders [115–117], the experiments demonstrated that the AT Transformer with a deep-shallow configuration (12 encoders pairing with 1 decoder) ran faster than the CMLM model of non-autoregressive decoding by a large margin when sentences per batch over 50 were processed. Further detail assumed that the computation was large enough to exhaust parallelism on the GPU in the scenario where batch processing was encountered, thus canceling out the speed benefit from NAD. Though aiming to speed up decoding, NAD may not run as fast as it seems. Specific circumstances such as hardware configuration, speed measurement, and application scenarios should be taken into consideration. Consequently, authentic fast decoding still earns little reward from practical applications and calls for more comprehensive and warranted work.

Investigate language setting Although nearly all aspects of translation speed and quality are emphasized, the application of NAMT to the configuration of diversified lingual orientations is rarely investigated and certainly valued. Initially, the original archetype of the NAMT model is drilled and validated in high-resource languages, with BLEU scores evaluated on official test sets, encompassing Newstest 2014 for WMT English-German, Newstest 2016 for WMT English-Romanian, or the development set for IWSLT16 English-German. When it comes to low-resource languages, however, training data-hungry NAT models is a non-trivial challenge, confronted with limited language processing tools and an inadequate parallel corpus of target minor languages, let alone potential accuracy degradation, which can be amplified due to the increased morphological complexity, such as Serbian [118,119], and the ulterior linguistic connections compared to resource-prosperous languages such as English and Romanian. To contend with this dilemma, extra data augmentation techniques such as back-translations offer viable means to train the NAT models, considering the inherently dual properties of machine translation tasks [120,121]. In addition, a combination of the Generative Adversarial Network [122] at the sentence level may also boost accuracy.

Additionally, varied language configurations such as monolingual, multilingual, and cross-lingual [123–125] wield certain impacts on the NAT models. More precisely, monolingual data are an integral part of the training of translation models, and the extensive use of target monolinguals can facilitate the fluency of output results. In the NAMT task, Sennrich et al. [120] procured a larger number of target monolingual corpora to train the NAT models, resulting in consistent performance advancements and stronger adaptabilities to long sentences. Moreover, Chi, ZW et al. [126] utilized a partially non-autoregressive generation pattern to predict that the remaining words belonged to the same span corruption in a multilingual text-to-text setting. This application yielded improved cross-lingual transferability over mT5 [127]. Agrawal et al. [128] adapted NAT models to multilingual scenarios, where six types of languages were involved. Though the dataset distilled from the multilingual teacher indeed showcased diminished lexical complexity and boosted alignment monotonicity, the mighty degenerated accuracy score manifested inevitably at an ill-suited time for multilingual NAT models.

7. Conclusions and Future Work

In this paper, we reviewed non-autoregressive neural machine translation during the past 5 years. Unlike a previous study that surveyed this domain from four aspects

at a fine granularity involving other non-autoregressive generation tasks, we paid more concentrated attention to non-autoregressive neural machine translation. Our study investigated this task from two new viewpoints at a coarse granularity, i.e., target dependency management and training strategy arrangement. According to the reported BLEU scores, quantitative graphs and qualitative comparisons were provided for various models to conduct a comprehensive analysis. Two prominent problems, target sentence length prediction and sequence-level knowledge distillation, were empirically observed for further inspection. Accumulative reexamination of translation quality and speedup suggested that non-autoregressive decoding may not run as fast as it seems and still lacks an authentic transcendence for accuracy. Based on that, potential work through the inner and outer facets of this task is also prospected.

In the near future, we call for more warrantable work as well as application-oriented research. Resolving internally knotty problems, such as target length prediction and sequence-level knowledge distillation, can lead to possible progress. While both become indispensable for the training of NAT models, they limit the model's performance to some extent. Determining target length in advance partially scaffolds output words yet augments indeterminacy in the output by identifying multiple candidates. Exposure bias between training and inference potentially aggravates accuracy errors. The combination of CTC and other stronger models to dynamically adapt target length may offer a possible way to elude this problem for accuracy promotion. Apart from that, distilled data with lower lexical diversity lack higher semantic information and are second-hand knowledge from AT teachers, inevitably leading to inferior quality. Training NAT models on raw data with GAN or other data augmentation techniques could possibly further promote the upper bound for translation quality.

Drawing external inspiration from similar domains or the emerging large language models may also help. More precisely, except for machine translation, sequence-to-sequence generation contains various tasks, including simultaneous translation (SMT), speech translation (ST), image caption (IC), automatic speech recognition (ASR), and text editing (TE). SMT and ASR require low response latency, which matches the notion of fast decoding in NAMT. ST and IC provide additional ideas for cross-modal fusion. TE is explicitly consistent with the paradigm of iterative decoding for fast generation. In addition, a flurry of recent work has been developed in the interest of large language models (LLM). When the parameter scale exceeds a certain level, the enlarged language models obtain surging performances and exhibit surprising capabilities that are unseen in smaller sizes, which are called emergent abilities. For example, the 175B-parameter GPT-3 shows extraordinary in-context learning ability yet is unavailable in the 1.5B-parameter GPT-2 and the 330M-parameter BERT. More remarkably, by adapting LLM (e.g., the GPT series) to practical applications such as dialogue, ChatGPT presents a striking conversational ability to interact with humans. Naturally, it may be a promising direction to adapt the powerful capacity of LLMs to a specific non-autoregressive translation task, leading to an interesting promotion.

Author Contributions: Conceptualization, X.Z.; methodology, F.L.; validation, J.C.; formal analysis, F.L.; investigation, F.L.; writing—original draft preparation, F.L.; writing—review and editing, F.L. and J.C.; supervision, X.Z.; project administration, X.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Key Projects of Guangxi Province, grant number 2020AA21077007; the Innovation Project of Guangxi Graduate Education, grant number YCSW2022042; the Guangxi New Engineering Research and Practice Project, grant number XGK2022003; and the Central Guidance on Local Science and Technology Development Fund of Guangxi Province, grant number 202201002. The APC was funded by the Science and Technology Key Projects of Guangxi Province, grant number 2020AA21077007.

Data Availability Statement: The data in this article are collected from previously published papers, where no new data were created. The BLEU scores in the table can be found in the corresponding quoted paper.

Acknowledgments: The authors would like to express their appreciation to the editors and reviewers for their valuable comments and suggestions. Relevant teachers from the School of Foreign Languages of Guangxi University are greatly appreciated for their valuable comments and suggestions, especially Zhaohui Bu. This work is supported by the Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi, China, and the School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, China.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
2. Cho, K.; Merriënboer, B.V.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
3. Bahdanau, D.; Cho, K.; Bengio, Y.J.C. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
4. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
5. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y. Convolutional Sequence to Sequence Learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
6. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the NIPS, Long Beach, CA, USA, 4–9 December 2017.
7. Radford, A.; Narasimhan, K. *Improving Language Understanding by Generative Pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
8. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models are Unsupervised Multitask Learners*; OpenAI: San Francisco, CA, USA, 2019.
9. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
10. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.J.A. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
11. Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genet. Program. Evolvable Mach.* **2018**, *19*, 305–307. [[CrossRef](#)]
12. Gu, J.; Bradbury, J.; Xiong, C.; Li, V.O.K.; Socher, R.J.A. Non-Autoregressive Neural Machine Translation. *arXiv* **2017**, arXiv:1711.02281.
13. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association-for-Computational-Linguistics, Univ Penn, Philadelphia, PA, USA, 7–12 July 2002.
14. Xiao, Y.; Wu, L.; Guo, J.; Li, J.; Zhang, M.; Qin, T.; Liu, T.-Y. A Survey on Non-Autoregressive Generation for Neural Machine Translation and Beyond. *arXiv* **2022**, arXiv:2204.09269. [[CrossRef](#)]
15. Han, H.; Indurthi, S.; Zaidi, M.A.; Lakumarapu, N.K.; Lee, B.; Kim, S.; Kim, C.; Hwang, I.J.A. Faster Re-translation Using Non-Autoregressive Model for Simultaneous Neural Machine Translation. *arXiv* **2020**, arXiv:2012.14681.
16. Han, H.; Ahn, S.; Choi, Y.; Chung, I.; Kim, S.; Cho, K. Monotonic Simultaneous Translation with Chunk-wise Reordering and Refinement. In Proceedings of the Conference on Machine Translation, online and in the Barceló Bávoro Convention Centre, Punta Cana, Dominican Republic, 7–11 November 2021.
17. Tian, Z.K.; Yi, J.Y.; Tao, J.H.; Bai, Y.; Zhang, S.; Wen, Z.Q. Spike-Triggered Non-Autoregressive Transformer for End-to-End Speech Recognition. In Proceedings of the Interspeech Conference, Shanghai, China, 25–29 October 2020.
18. Fujita, Y.; Watanabe, S.; Omachi, M.; Chang, X.K. Insertion-Based Modeling for End-to-End Automatic Speech Recognition. In Proceedings of the Interspeech Conference, Shanghai, China, 25–29 October 2020.
19. Leng, Y.C.; Tan, X.; Zhu, L.C.; Xu, J.; Luo, R.Q.; Liu, L.Q.; Qin, T.; Li, X.Y.; Lin, E.; Liu, T.Y. FastCorrect: Fast Error Correction with Edit Alignment for Automatic Speech Recognition. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS), Online, 6–14 December 2021.
20. Inaguma, H.; Kawahara, T.; Watanabe, S. Source and Target Bidirectional Knowledge Distillation for End-to-end Speech Translation. In Proceedings of the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics-Human Language Technologies (NAACL-HLT), Online, 6–11 June 2021.
21. Inaguma, H.; Dalmia, S.; Yan, B.; Watanabe, S. FAST-MD: Fast Multi-Decoder End-To-End Speech Translation with Non-Autoregressive Hidden Intermediates. In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021.
22. Tokarchuk, E.; Rosendahl, J.; Wang, W.Y.; Petrushkov, P.; Lancewicki, T.; Khadivi, S.; Ney, H. Integrated Training for Sequence-to-Sequence Models Using Non-Autoregressive Transformer. In Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT), Online, 5–6 August 2021.

23. Guo, L.T.; Liu, J.; Zhu, X.X.; He, X.J.; Jiang, J.; Lu, H. Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021.
24. Mallinson, J.; Severyn, A.; Malmi, E.; Garrido, G.J.A. FELIX: Flexible Text Editing Through Tagging and Insertion. *arXiv* **2020**, arXiv:2003.10687.
25. Wan, D.; Kedzie, C.; Ladhak, F.; Carpuat, M.; McKeown, K. Incorporating Terminology Constraints in Automatic Post-Editing. In Proceedings of the Conference on Machine Translation, Online, 19–20 November 2020.
26. Xu, W.J.; Carpuat, M. EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 311–328. [[CrossRef](#)]
27. Agrawal, S.; Carpuat, M. An Imitation Learning Curriculum for Text Editing with Non-Autoregressive Models. In Proceedings of the 60th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Dublin, Ireland, 22–27 May 2022.
28. Niwa, A.; Takase, S.; Okazaki, N.J.J.I.P. Nearest Neighbor Non-autoregressive Text Generation. *J. Inf. Process.* **2022**, *31*, 344–352. [[CrossRef](#)]
29. Xu, J.; Crego, J.M.; Yvon, F. Bilingual Synchronization: Restoring Translational Relationships with Editing Operations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, UAE, 7–11 December 2022.
30. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
31. ArXiv, O.J. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
32. Weaver, W. *Machine Translation of Languages: Fourteen Essays*; Locke, W.N., Booth, A.D., Eds.; Technology Press of the Massachusetts Institute of Technology: Cambridge, MA, USA; John Wiley & Sons, Inc.: New York, NY, USA, 1955; pp. 15–23.
33. Koehn, P.; Och, F.J.; Marcu, D. Statistical Phrase-Based Translation. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Edmonton, Canada, 27 May–1 June 2003.
34. Sánchez-Martínez, F.; Pérez-Ortiz, J.A. Philipp Koehn, Statistical machine translation. *Mach. Transl.* **2010**, *24*, 273–278. [[CrossRef](#)]
35. Kaiser, L.; Roy, A.; Vaswani, A.; Parmar, N.; Bengio, S.; Uszkoreit, J.; Shazeer, N. Fast Decoding in Sequence Models Using Discrete Latent Variables. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
36. Oord, A.v.d.; Vinyals, O.; Kavukcuoglu, K.J.A. Neural Discrete Representation Learning. *arXiv* **2017**, arXiv:1711.00937.
37. Kaiser, L.; Bengio, S.J.A. Discrete Autoencoders for Sequence Models. *arXiv* **2018**, arXiv:1801.09797.
38. Roy, A.; Vaswani, A.; Neelakantan, A.; Parmar, N.J.A. Theory and Experiments on Vector Quantized Autoencoders. *arXiv* **2018**, arXiv:1805.11063.
39. Ma, X.Z.; Zhou, C.T.; Li, X.; Neubig, G.; Hovy, E. FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow. In Proceedings of the Conference on Empirical Methods in Natural Language Processing/9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019.
40. Song, J.; Kim, S.; Yoon, S. AligNART: Non-autoregressive Neural Machine Translation by Jointly Learning to Estimate Alignment and Translate. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Punta Cana, Dominican Republic, 7–11 November 2021.
41. Heo, D.; Choi, H.J.A. Shared Latent Space by Both Languages in Non-Autoregressive Neural Machine Translation. *arXiv* **2023**, arXiv:2305.03511.
42. Akoury, N.; Krishna, K.; Iyyer, M. Syntactically Supervised Transformers for Faster Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
43. Liu, Y.; Wan, Y.; Zhang, J.G.; Zhao, W.T.; Yu, P.S. 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 (EACL), Kyiv, Ukraine, 19–23 April 2021.
44. Bao, Y.; Huang, S.J.; Xiao, T.; Wang, D.Q.; Dai, X.Y.; Chen, J.J. Non-Autoregressive Translation by Learning Target Categorical Codes. In Proceedings of the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics—Human Language Technologies (NAACL-HLT), Online, 6–11 June 2021.
45. Ran, Q.; Lin, Y.K.; Li, P.; Zhou, J. Guiding Non-Autoregressive Neural Machine Translation Decoding with Reordering Information. In Proceedings of the 35th AAAI Conference on Artificial Intelligence/33rd Conference on Innovative Applications of Artificial Intelligence/11th Symposium on Educational Advances in Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021.
46. Bao, Y.; Zhou, H.; Feng, J.; Wang, M.; Huang, S.; Chen, J.; Lei, L.J.A. Non-autoregressive Transformer by Position Learning. *arXiv* **2019**, arXiv:1911.10677.
47. Lee, J.; Mansimov, E.; Cho, K. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018.
48. Shu, R.; Lee, J.; Nakayama, H.; Cho, K. Latent-Variable Non-Autoregressive Neural Machine Translation with Deterministic Inference Using a Delta Posterior. In Proceedings of the 34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020.

49. Lee, J.; Shu, R.; Cho, K. Iterative Refinement in the Continuous Space for Non-Autoregressive Neural Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020.
50. Wang, C.Q.; Zhang, J.; Chen, H.Q. Semi-Autoregressive Neural Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018.
51. Stern, M.; Chan, W.; Kiros, J.; Uszkoreit, J. Insertion Transformer: Flexible Sequence Generation via Insertion Operations. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.
52. Kasai, J.; Cross, J.; Ghazvininejad, M.; Gu, J.T. Non-autoregressive Machine Translation with Disentangled Context Transformer. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 12–18 July 2020.
53. Ran, Q.; Lin, Y.K.; Li, P.; Zhou, J. 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 (ACL), Online, 5–10 July 2020.
54. Guo, P.; Xiao, Y.; Li, J.; Zhang, M.J.A. RenewNAT: Renewing Potential Translation for Non-Autoregressive Transformer. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
55. Ghazvininejad, M.; Levy, O.; Liu, Y.H.; Zettlemoyer, L. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing/9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019.
56. Kreuzer, J.; Foster, G.; Cherry, C. Inference Strategies for Machine Translation with Conditional Masking. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020.
57. Xiao, Y.; Xu, R.; Wu, L.; Li, J.; Qin, T.; Liu, Y.-T.; Zhang, M.J.A. AMOM: Adaptive Masking over Masking for Conditional Masked Language Model. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
58. Libovicky, J.; Helcl, J. End-to-End Non-Autoregressive Neural Machine Translation with Connectionist Temporal Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018.
59. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; Association for Computing Machinery: Pittsburgh, PA, USA, 2006; pp. 369–376.
60. Wang, Y.R.; Tian, F.; He, D.; Qin, T.; Zhai, C.X.; Liu, T.Y. Non-Autoregressive Machine Translation with Auxiliary Regularization. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence/31st Innovative Applications of Artificial Intelligence Conference/9th AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
61. Li, Z.H.; Lin, Z.; He, D.; Tian, F.; Qin, T.; Wang, L.W.; Liu, T.Y. Hint-Based Training for Non-Autoregressive Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing/9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019.
62. Shao, C.Z.; Zhang, J.C.; Feng, Y.; Meng, F.D.; Zhou, J. Minimizing the Bag-of-Ngrams Difference for Non-Autoregressive Neural Machine Translation. In Proceedings of the 34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
63. Ghazvininejad, M.; Karpukhin, V.; Zettlemoyer, L.; Levy, O. Aligned Cross Entropy for Non-Autoregressive Machine Translation. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 12–18 July 2020.
64. Tu, L.F.; Pang, R.Y.Z.; Wiseman, S.; Gimpel, K. ENGINE: Energy-Based Inference Networks for Non-Autoregressive Machine Translation. In Proceedings of the 58th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Online, 5–10 July 2020.
65. Du, C.X.; Tu, Z.P.; Jiang, J. Order-Agnostic Cross Entropy for Non-Autoregressive Machine Translation. In Proceedings of the International Conference on Machine Learning (ICML), Online, 18–24 July 2021.
66. Shao, C.Z.; Feng, Y.; Zhang, J.C.; Meng, F.D.; Zhou, J. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. *Comput. Linguist.* **2021**, *47*, 891–925. [[CrossRef](#)]
67. Shao, C.Z.; Feng, Y.; Zhang, J.C.; Meng, F.D.; Chen, X.L.; Zhou, J. Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
68. Wei, B.Z.; Wang, M.X.; Zhou, H.; Lin, J.Y.; Sun, X. Imitation Learning for Non-Autoregressive Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
69. Sun, Z.Q.; Li, Z.H.; Wang, H.Q.; He, D.; Lin, Z.; Deng, Z.H. Fast Structured Decoding for Sequence Models. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
70. Kasner, Z.; Libovický, J.; Helcl, J.J.A. Improving Fluency of Non-Autoregressive Machine Translation. *arXiv* **2020**, arXiv:2004.03227.
71. Guo, L.; Liu, J.; Zhu, X.; Lu, H.J.A. Fast Sequence Generation with Multi-Agent Reinforcement Learning. *arXiv* **2021**, arXiv:2101.09698.
72. Shan, Y.; Feng, Y.; Shao, C.Z. Modeling Coverage for Non-Autoregressive Neural Machine Translation. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Online, 18–22 July 2021.

73. Guo, J.L.; Tan, X.; Xu, L.L.; Qin, T.; Chen, E.H.; Liu, T.Y. Fine-Tuning by Curriculum Learning for Non-Autoregressive Neural Machine Translation. In Proceedings of the 34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
74. Liu, J.L.; Ren, Y.; Tan, X.; Zhang, C.; Qin, T.; Zhao, Z.; Liu, T.Y. Task-Level Curriculum Learning for Non-Autoregressive Neural Machine Translation. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021.
75. Guo, J.; Zhang, Z.; Xu, L.; Wei, H.-R.; Chen, B.; Chen, E.J.A. Incorporating BERT into Parallel Sequence Decoding with Adapters. *arXiv* **2020**, arXiv:2010.06138.
76. Su, Y.X.; Cai, D.; Wang, Y.; Vandyke, D.; Baker, S.; Li, P.J.; Collier, N. Non-Autoregressive Text Generation with Pre-trained Language Models. In Proceedings of the 16th Conference of the European-Chapter-of-the-Association-for-Computational-Linguistics (EACL), Online, 19–23 April 2021.
77. Li, P.F.; Li, L.Y.; Zhang, M.; Wu, M.H.; Liu, Q. Universal Conditional Masked Language Pre-training for Neural Machine Translation. In Proceedings of the 60th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Dublin, Ireland, 22–27 May 2022.
78. Guo, J.L.; Tan, X.; He, D.; Qin, T.; Xu, L.L.; Liu, T.Y. Non-Autoregressive Neural Machine Translation with Enhanced Decoder Input. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
79. Gu, J.; Kong, X.J.A. Fully Non-autoregressive Neural Machine Translation: Tricks of the Trade. *arXiv* **2020**, arXiv:2012.15833.
80. Huang, F.; Zhou, H.; Liu, Y.; Li, H.; Huang, M.J.A. Directed Acyclic Transformer for Non-Autoregressive Machine Translation. In Proceedings of the 39th International Conference on Machine Learning (ICML 2022), Baltimore, MD, USA, 17–23 July 2022.
81. Shao, C.; Ma, Z.; Feng, Y.J.A. Viterbi Decoding of Directed Acyclic Transformer for Non-Autoregressive Machine Translation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022.
82. Ma, Z.; Shao, C.; Gui, S.; Zhang, M.; Feng, Y.J.A. Fuzzy Alignments in Directed Acyclic Graph for Non-Autoregressive Machine Translation. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR2023), Kigali, Rwanda, 1–5 May 2023.
83. Huang, F.; Ke, P.; Huang, M.J.A. Directed Acyclic Transformer Pre-training for High-quality Non-autoregressive Text Generation. *arXiv* **2023**, arXiv:2304.11791.
84. Shao, C.; Zhang, J.; Zhou, J.; Feng, Y.J.A. Rephrasing the Reference for Non-Autoregressive Machine Translation. In Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI2023), Washington, DC, USA, 7–14 February 2023.
85. Qian, L.H.; Zhou, H.; Bao, Y.; Wang, M.X.; Qiu, L.; Zhang, W.N.; Yu, Y.; Li, L. Glancing Transformer for Non-Autoregressive Neural Machine Translation. In Proceedings of the Joint Conference of 59th Annual Meeting of the Association-for-Computational-Linguistics (ACL)/11th International Joint Conference on Natural Language Processing (IJCNLP)/6th Workshop on Representation Learning for NLP (RepL4NLP), Bangkok, Thailand, 1–6 August 2021.
86. Xie, P.; Li, Z.; Hu, X.J.A. MvSR-NAT: Multi-view Subset Regularization for Non-Autoregressive Machine Translation. *arXiv* **2021**, arXiv:2108.08447. [[CrossRef](#)]
87. Savinov, N.; Chung, J.; Binkowski, M.; Elsen, E.; Oord, A.v.d.J.A. Step-unrolled Denoising Autoencoders for Text Generation. In Proceedings of the Tenth International Conference on Learning Representations (ICLR 2022), Online, 25–29 April 2022.
88. Huang, C.Y.; Zhou, H.; Zaiane, O.R.; Mou, L.L.; Li, L. Non-autoregressive Translation with Layer-Wise Prediction and Deep Supervision. In Proceedings of the 36th AAAI Conference on Artificial Intelligence/34th Conference on Innovative Applications of Artificial Intelligence/12th Symposium on Educational Advances in Artificial Intelligence, Online, 22 February–1 March 2022.
89. Wang, M.; Guo, J.; Wang, Y.; Wei, D.; Shang, H.; Su, C.; Chen, Y.; Li, Y.; Zhang, M.; Tao, S.; et al. Diformer: Directional Transformer for Neural Machine Translation. In Proceedings of the European Association for Machine Translations Conferences/Workshops, Ghent, Belgium, 1–3 June 2022.
90. Ge, T.; Xia, H.; Sun, X.; Chen, S.-Q.; Wei, F.J.A. Lossless Acceleration for Seq2seq Generation with Aggressive Decoding. *arXiv* **2022**, arXiv:2205.10350.
91. Qin, B.; Jia, A.; Wang, Q.; Lu, J.; Pan, S.; Wang, H.; Chen, M. The RoyalFlush System for the WMT 2022 Efficiency Task. In Proceedings of the EMNLP 2022 Seventh Conference on Machine Translation, Abu Dhabi, United Arab Emirates, 7–11 December 2022.
92. Wang, X.; Zheng, Z.; Huang, S. Helping the Weak Makes You Strong: Simple Multi-Task Learning Improves Non-Autoregressive Translators. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2022), Abu Dhabi, United Arab Emirates, 7–11 December 2022.
93. Saharia, C.; Chan, W.; Saxena, S.; Norouzi, M. Non-Autoregressive Machine Translation with Latent Alignments. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), Online, 16–20 November 2020.
94. Chan, W.; Saharia, C.; Hinton, G.; Norouzi, M.; Jaitly, N. Imputer: Sequence Modelling via Imputation and Dynamic Programming. In Proceedings of the 25th Americas Conference on Information Systems of the Association-for-Information-Systems (AMCIS 2019), Cancun, Mexico, 15–17 August 2019.
95. Zhang, K.X.; Wang, R.; Tan, X.; Guo, J.L.; Ren, Y.; Qin, T.; Liu, T.Y. A Study of Syntactic Multi-Modality in Non-Autoregressive Machine Translation. In Proceedings of the Conference of the North-American-Chapter-of-the-Association-for-Computational-Linguistics (NAACL)—Human Language Technologies, Seattle, WA, USA, 10–15 July 2022.

96. Shao, C.; Feng, Y.J.A. Non-Monotonic Latent Alignments for CTC-Based Non-Autoregressive Machine Translation. In Proceedings of the Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans Convention Center, New Orleans, LA, USA, 28 November–9 December 2022.
97. Du, C.; Tu, Z.; Jiang, J.J.A. ngram-OAXE: Phrase-Based Order-Agnostic Cross Entropy for Non-Autoregressive Machine Translation. In Proceedings of the 29th International Conference on Computational Linguistics (Coling 2022 Oral), Gyeongju, Republic of Korea, 12–17 October 2022.
98. Li, Y.; Cui, L.; Yin, Y.; Zhang, Y. Multi-Granularity Optimization for Non-Autoregressive Translation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP2022), Abu Dhabi, United Arab Emirates, 7–11 December 2022.
99. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
100. Sun, Z.; Yang, Y. An EM Approach to Non-autoregressive Conditional Sequence Generation. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020.
101. Cho, K.J.A. Noisy Parallel Approximate Decoding for Conditional Recurrent Language Model. *arXiv* **2016**, arXiv:1605.03835.
102. Kim, Y.; Rush, A.M.J.A. Sequence-Level Knowledge Distillation. In Proceedings of the EMNLP 2016, Austin, Texas, USA, 1–5 November 2016.
103. Hinton, G.E.; Vinyals, O.; Dean, J.J.A. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
104. Zhou, C.; Neubig, G.; Gu, J.J.A. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
105. Xu, W.; Ma, S.; Zhang, D.; Carpuat, M. How Does Distilled Data Complexity Impact the Quality and Confidence of Non-Autoregressive Machine Translation? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2021), Online, 1–6 August 2021.
106. Ren, Y.; Liu, J.L.; Tan, X.; Zhao, Z.; Zhao, S.; Liu, T.Y. A Study of Non-autoregressive Model for Sequence Generation. In Proceedings of the 58th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Online, 5–10 July 2020.
107. Zhou, J.W.; Keung, P.; Assoc Computat, L. Assoc Computat. Improving Non-autoregressive Neural Machine Translation with Monolingual Data. In Proceedings of the 58th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Online, 5–10 July 2020.
108. Guo, J.; Wang, M.; Wei, D.; Shang, H.; Wang, Y.; Li, Z.; Yu, Z.; Wu, Z.; Chen, Y.; Su, C.; et al. Self-Distillation Mixup Training for Non-autoregressive Neural Machine Translation. *arXiv* **2021**, arXiv:2112.11640.
109. Shao, C.; Wu, X.; Feng, Y. One Reference Is Not Enough: Diverse Distillation with Reference Selection for Non-Autoregressive Translation. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Seattle, WA, USA, 10–15 July 2022.
110. Liu, M.; Bao, Y.; Zhao, C.; Huang, S.J.A. Selective Knowledge Distillation for Non-Autoregressive Neural Machine Translation. *arXiv* **2023**, arXiv:2303.17910. [[CrossRef](#)]
111. Binkowski, M.; Donahue, J.; Dieleman, S.; Clark, A.; Elsen, E.; Casagrande, N.; Cobo, L.C.; Simonyan, K.J.A. High Fidelity Speech Synthesis with Adversarial Networks. *arXiv* **2019**, arXiv:1909.11646.
112. Kasai, J.; Pappas, N.; Peng, H.; Cross, J.; Smith, N.A. Deep Encoder, Shallow Decoder: Reevaluating Non-autoregressive Machine Translation. In Proceedings of the International Conference on Learning Representations, Online, 26 April–1 May 2020.
113. Helcl, J.; Haddow, B.; Birch, A.J.A. Non-Autoregressive Machine Translation: It's Not as Fast as it Seems. *arXiv* **2022**, arXiv:2205.01966.
114. Schmidt, R.M.; Pires, T.; Peitz, S.; Löff, J.J.A. Non-Autoregressive Neural Machine Translation: A Call for Clarity. *arXiv* **2022**, arXiv:2205.10577.
115. Barone, A.V.M.; Helcl, J.; Sennrich, R.; Haddow, B.; Birch, A.J.A. Deep architectures for Neural Machine Translation. In Proceedings of the WMT 2017 Research Track, Copenhagen, Denmark, 7–8 September 2017.
116. Wang, Q.; Li, B.; Xiao, T.; Zhu, J.B.; Li, C.L.; Wong, D.F.; Chao, L.S. Learning Deep Transformer Models for Machine Translation. In Proceedings of the 57th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
117. Kim, Y.J.; Junczys-Dowmunt, M.; Hassan, H.; Heafield, K.; Grundkiewicz, R.; Bogoychev, N. From Research to Production and Back: Ludicrously Fast Neural Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 4 November 2019.
118. Batanović, V.; Cvetanović, M.; Nikolić, B.J.P.O. A versatile framework for resource-limited sentiment articulation, annotation, and analysis of short texts. *PLoS ONE* **2020**, *15*, e0242050. [[CrossRef](#)] [[PubMed](#)]
119. Draskovic, D.; Zecevic, D.; Nikolic, B. Development of a Multilingual Model for Machine Sentiment Analysis in the Serbian Language. *Mathematics* **2022**, *10*, 3236. [[CrossRef](#)]
120. Sennrich, R.; Haddow, B.; Birch, A.J.A. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany; pp. 86–96.

121. He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.-Y.; Ma, W.-Y. Dual learning for machine translation. In Proceedings of the 30th International Conference on Neural Information Processing Systems; Curran Associates Inc.: Barcelona, Spain, 2016; pp. 820–828.
122. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *J. Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
123. Kostić, M.; Batanović, V.; Nikolić, B. Monolingual, multilingual and cross-lingual code comment classification. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106485. [[CrossRef](#)]
124. Zhu, Y.; Feng, J.; Zhao, C.; Wang, M.; Li, L. Counter-Interference Adapter for Multilingual Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 November 2021.
125. Liu, Y.H.; Gu, J.T.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 726–742. [[CrossRef](#)]
126. Chi, Z.W.; Dong, L.; Ma, S.M.; Huang, S.H.; Mao, X.L.; Huang, H.Y.; Wei, F.R. mT6: Multilingual Pretrained Text-to-Text Transformer with Translation Pairs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online and Punta Cana, Dominican Republic, 7–11 November 2021.
127. Xue, L.T.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics—Human Language Technologies (NAACL-HLT), Online, 6–11 June 2021.
128. Agrawal, S.; Kreutzer, J.; Cherry, C.J.A. Can Multilinguality benefit Non-autoregressive Machine Translation? *arXiv* **2021**, arXiv:2112.08570.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.