applied sciences

MDPI

Article

Improving Non-Autoregressive Machine Translation Using Sentence-Level Semantic Agreement

Shuheng Wang [1], Heyan Huang [2] and Shumin Shi [2,*]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; wsh@njust.edu.cn
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100811, China; hhy63@bit.edu.cn
* Correspondence: bjssm@bit.edu.cn

Abstract: Theinference stage can be accelerated significantly using a Non-Autoregressive Transformer (NAT). However, the training objective used in the NAT model also aims to minimize the loss between the generated words and the golden words in the reference. Since the dependencies between the target words are lacking, this training objective computed at word level can easily cause semantic inconsistency between the generated and source sentences. To alleviate this issue, we propose a new method, Sentence-Level Semantic Agreement (SLSA), to obtain consistency between the source and generated sentences. Specifically, we utilize contrastive learning to pull the sentence representations of the source and generated sentences closer together. In addition, to strengthen the capability of the encoder, we also integrate an agreement module into the encoder to obtain a better representation of the source sentence. The experiments are conducted on three translation datasets: the WMT 2014 EN → DE task, the WMT 2016 EN → RO task, and the IWSLT 2014 DE → DE task, and the improvement in the NAT model's performance shows the effect of our proposed method.

Keywords: machine translation; non-autoregressive; contrastive learning; semantic agreement

check for updates

Citation: Shuheng, W.; Heyan, H.; Shumin, S. Improving Non-Autoregressive Machine Translation Using Sentence-Level Semantic Agreement. *Appl. Sci.* **2022**, *12*, 5003. https://doi.org/10.3390/app12105003

Academic Editors: Phivos Mylonas, Katia Lida Kermanidis and Manolis Maragoudakis

Received: 19 April 2022
Accepted: 13 May 2022
Published: 16 May 2022

1. Introduction

Over the years, tremendous success has been achieved in encoder–decoder based neural machine translation (NMT) [1–3]. The encoder maps the source sentence into a hidden representation, and the target sentence is generated by the decoder from the hidden representation in an autoregressive method. This autoregressive method has assisted the NMT model in obtaining high accuracy [3]. However, because it needs the previously predicted words as inputs, this also limits the speed of the inference stage. Recently, Gu et al. [4] proposed a non-autoregressive transformer (NAT) to break the limitation and reduce the inference latency. In general, the NAT model also utilizes the encoder–decoder framework. However, by removing the autoregressive method in the decoder, the NAT model can significantly expedite the decoding stage. Yet, the performance of the NAT model still lags behind the NMT model.

During training, the NAT model, as the NMT model, uses a word-level cross-entropy to optimize the whole model. Nevertheless, under the background of non-autoregressive translation, the dependencies in the target words cannot be learned properly with the word-level cross-entropy [5]. Although it encourages the NAT model to generate the correct token at each position, due to the lack of target dependency, the NAT model cannot consider global correctness. The NAT model cannot efficiently model the target dependency well, and the cross-entropy loss further weakens this feature, causing undertranslation or overtranslation [5]. Recently, some research has proposed ways to alleviate this issue. For example, Sun et al. [6] utilized a CRF module to model the global path in the decoder, and Shao et al. [5] used a bag-of-words loss to encourage the NAT model to capture the target dependency. However, this previous research only considered global or partial modeling of dependency on the target side. Another issue that cannot be ignored is that the semantics of the generated sentence cannot be guaranteed to be consistent with the source sentence.

In contrast, in human translation, the translator translates a sentence by its sentence meaning, instead of the word-by-word meaning. Inspired by this process, the Sentence-Level Semantic Agreement (SLSA) method is proposed in this paper to shorten the distance between the source and generated sentence representations. SLSA utilizes contrastive learning to ensure the semantic consistency between the source and generated sentences. In addition, SLSA also adapts contrastive learning in the encoder to ensure the encoder correctly transforms the source sentence into a shared representation space, which enables the decoder to extract the information from it more easily.

The performance of the SLSA is evaluated by experiments on three translation benchmarks, the WMT 2014 EN → DE task, the WMT 2016 EN → RO task, and the IWSLT 2014 DE → EN task. The results indicate that a significant improvement in the NAT model is achieved via our proposed method.

The remainder of this paper is structured as follows: Section 2 describes the background and baseline model of our work. Section 3 describes our proposed method in detail. In Section 4, we describe the conducted experiments and analyze the results. Section 5 provides an overview of related works. Finally, we conclude our work and present an outlook for future research.

2. Background

2.1. Non-Autoregressive Translation

By generating the target words in one shot, Non-Autoregressive Translation [4] is utilized to speed up the inference stage. The vanilla NAT model adapts a similar encoder–decoder framework to that used in autoregressive translation. Furthermore, the encoder used in the NAT model remains the same as the transformer [3]. Different from the autoregressive translation, the NAT model utilizes a bidirectional mask in the decoder, on which the non-autoregressive mechanism mainly relies. In addition, because the NAT model is unable to decide the length of a target sequence, as the autoregressive translation does effectively, we use a separate predictor to output the length.

Using $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ to denote the source and corresponding target sentences, the NAT model models the target sentence as an independent conditional probability:

$$P(Y|X) = \prod_{t=1}^{N} p(y_t|X, \theta). \tag{1}$$

In this way, the probability of a word at each position only depends on the representation of the source sequence, and the parameter θ is learned with the cross-entropy by minimizing the negative log-likelihood as:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p(y_t|X; \theta). \tag{2}$$

During inference, the word with the maximum probability at each position forms the final translation:

$$\hat{y}_t = arg\max p(y_t|X; \theta). \tag{3}$$

2.2. Glancing Transformer

Although the vanilla NAT model can significantly reduce the decoding latency, the translation accuracy is much lower than that of the autoregressive translation. To improve the performance, Qian et al. [7] introduced a *Glancing Transformer* (GLAT). With GLAT, the gap between the NAT and AT models is narrowed.

The training process of GLAT can be divided into two steps. Firstly, GLAT copies the input from the source embeddings, feeds it into the decoder, and generates the target sentence. Secondly, GLAT computes the distance between the generated target sentence and the reference sentence. According to the distance, GLAT samples some words from the reference sentence:

$$\mathbb{GS} = S(Y, D(Y, \hat{Y})). \tag{4}$$

Then the sampled words are combined with the input at the first step. The new input is then fed into the decoder to perform the second pass. Different from the vanilla NAT model, GLAT does not compute the loss at each position, but it computes the loss at the position that is not sampled:

$$\mathcal{L}_{GT}(\theta) = \sum_{y \notin \mathbb{GS}} \log p(y_t | \mathbb{GS}, X; \theta) \tag{5}$$

Although the Glancing Transformer can capture the latent target dependency and improve the performance of the NAT model, it still uses cross-entropy to train the whole model and cannot guarantee the semantic agreement between the source and target sentences. In other words, GLAT also faces the same problem: the semantics of the generated sentence are not consistent with the source sentence.

3. Method

Previous work [8,9] has pointed out the improvements in word alignment, especially how it can improve the performance of the NAT model. However, the previous work is mainly based on the word-level semantic alignment. Meanwhile, the effect of self-attention [3] is to find the nested words in the source sentence for the word in the target sentence. From the model to the training objective, there is no mechanism to ensure the semantic alignment between the source and target sentence. In this paper, we attempt to explore the sentence-level semantic relationship between the source and target sentences. We propose a novel method, Sentence-Level Semantic Agreement (SLSA), to ensure the representation of the source and generated sentences is similar.

First, we need to obtain the sentence-level representation of the source and generated sentences. As pointed out in previous work [10,11], the average of the word-level representations is a effective way to represent a sentence. Similarly, in machine translation, it is also a good representation for a sentence [12,13]. So in this work, we also utilize this method to represent a sentence.

Given a source sentence, we use $H_E = \{h_E^1, h_E^2, \ldots, h_E^m\}$ as the word-level representations output by the top layer of an encoder. Similarly, $H_D = \{h_D^1, h_D^2, \ldots, h_D^n\}$ are the word-level representations output by the top layer of a decoder. The source sentence-level representation $\overline{H}_E$ is computed as:

$$\overline{H}_E = \frac{1}{m} \sum_{t=1}^{m} h_E^t. \tag{6}$$

The generated sentence-level representation $\overline{H}_D$ is computed by the same method:

$$\overline{H}_D = \frac{1}{n} \sum_{t=1}^{n} h_D^t. \tag{7}$$

3.1. Semantic Agreement between Source and Generated Sentence

According to the denotation above, the key to the semantic agreement between the source and generated sentences is minimizing the distance between $\overline{H}_E$ and $\overline{H}_D$. In this work, the SLSA uses contrastive loss to bring the representations of the source and generated sentences close together (Figure 1a) and keep the semantic representations generated by the encoder and decoder consistent.

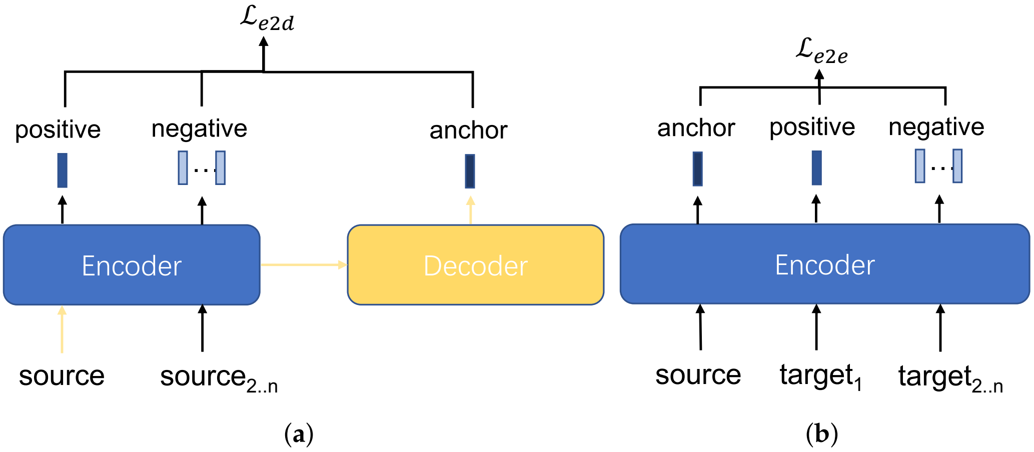


Figure 1. The semantic agreement in our model. (**a**) is the semantic agreement between the source and generated sentences. It uses the generated and source sentences as the positive pair and the generated and randomly sampled source sentences as the negative pair. (**b**) is the semantic agreement in the encoder. Different from (**a**), it only uses the representation output by the encoder. Furthermore, it utilizes the source and corresponding target sentences as the positive pair and the source and randomly sampled target sentence as the negative pair. Our model computes the contrastive loss $\mathcal{L}_{e2d}$ and $\mathcal{L}_{e2e}$ on the representation of the positive and negative pairs.

The idea of contrastive loss is to minimize the distance between relevant sentences and maximum the distance between irrelevant sentences. In this section, contrastive loss is utilized to increase the similarity between the source and generated sentences. Given a bilingual sentence pair $(X_i, \hat{Y}_i)$, which denotes the positive example, a sentence X_j $(i \neq j)$ is randomly selected to form the negative example $(X_j, \hat{Y}_i)$. In this work, to simplify the implementation, we sample the negative examples from the same batch. We compute the contrastive loss as:

$$\mathcal{L}_{e2d} = -\log \frac{\exp(\mathrm{sim}(\overline{H}_E(X_i), \overline{H}_D(\hat{Y}_i)))/\tau}{\sum_j \exp(\mathrm{sim}(\overline{H}_E(X_j), \overline{H}_D(\hat{Y}_i)))/\tau}, \tag{8}$$

in which $\mathrm{sim}(\cdot)$ denotes the similarity between two sentences. The difficulty of distinguishing between the positive and negative examples is controlled by τ. In this way, contrastive loss can pull the semantic representations of source and generated sentences close together. The $softmax$ function pushes the representations of irrelevant sentences away from each other.

3.2. Semantic Agreement in Encoder

The conclusion in [13] showed that mapping the representations of the source and generated sentences by the encoder into the shared space can lead to a better translation. So, we also introduced a semantic agreement loss into the encoder (Figure 1b) to map the representations of the source sentence and the target sentence into a shared space.

Similar to the description in Section 3.1, we denote $(X_i, Y_i)(Y_i$ is different from $\hat{Y}_i$. Y_i is sampled from the reference, and $\hat{Y}_i$ is the sentence generated by the decoder) as the positive sample and (X_i, Y_j) as the negative sample, respectively. We feed them into the encoder and compute the semantic agreement loss as:

$$\mathcal{L}_{e2e} = -\log \frac{\exp(\mathrm{sim}(\overline{H}_E(X_i), \overline{H}_E(Y_i)))/\tau}{\sum_j \exp(\mathrm{sim}(\overline{H}_E(X_i), \overline{H}_E(Y_j)))/\tau}. \tag{9}$$

In this way, we can force the semantic representation of the source sentence learned by the encoder to be closer to the semantics of the target sentence, which may reduce the difficulty in generating the target sentence.

3.3. Ahead Supervision for Better Representation

In a vanilla NAT model, the input to the decoder is copied from the encoder to remove the dependency on the previously generated words. Furthermore, the supervised signal is only on the output of last layer. In this way, the sentence-level semantic representation output by the decoder may not be correct. Meanwhile, the similarity between the representations of the source and generated sentences is small, but the loss $\mathcal{L}_{e2d}$ is very large, which may cause the model to try to reduce the distance between the two representations and fail to learn the correct translation.

In this work, to obtain a better semantic representation, we added a supervised signal before the last layer [14]. We computed the cross-entropy loss on the output of the penultimate layer:

$$\mathcal{L}_{A_{GT}}(\theta) = \sum_{y \notin \mathbb{GS}} \log p(y_t | \mathbb{GS}, X; \theta). \quad (10)$$

In this way, we ensured that the semantic representations obtained from the last layer were correct.

3.4. Optimization and Inference

In this work, given the performance of the Glancing Transformer, we adopted the Glancing Transformer as the base of our method and replaced the attention-based *Soft Copy* with *Uniform Copy*. The parameters of whole model were jointly learned by minimizing the loss $\mathcal{L}$, which is the sum of Equations (5), (9), and (10):

$$\mathcal{L} = \mathcal{L}_{GT} + \mathcal{L}_{e2d} + \mathcal{L}_{e2e} + \mathcal{L}_{A_{GT}}. \quad (11)$$

During inference, because our model, SLSA, only modifies the training procedure, it can perform a vanilla decoding pass.

4. Experiments

In this section, the conducted experiments are detailed to show the performance of our method.

4.1. Datasets and Settings

4.1.1. Datasets

For a fair comparison, we followed previous works [4,7] and conducted the experiments on the benchmark tasks: WMT 2014 EN → DE, WMT 2016 EN → RO, and IWSLT 2014 DE → EN. The training datasets of these tasks contain 4.5 M, 610 k, and 190 k bilingual sentence pairs, respectively. After we preprocessed these datasets following the preprocessing steps in [15], each word in the dataset was divided into the sub-word units using BPE [16]. For WMT 2014 EN → DE, newstest-2013 and newstest-2014 were used as the development and test sets, respectively. For WMT 2016 EN → RO, newsdev-2016 and newstest-2016 were employed as development and test sets, respectively. Furthermore, for IWSLT 2014 DE → EN, we merged dev2010, dev2012, test2010, test2011, and test2012 together as the test set.

4.1.2. Sequence-Level Knowledge Distillation

As pointed out in previous works [4,7,15,17], knowledge distillation is a critical technology for non-autoregressive translation. In this work, we distilled the train set [18] for all tasks. We used the transformer with the base setting [3] to generate the distilled datasets. Then, all our models were trained on the distilled datasets.

4.1.3. Baselines

To show the performance of our method, we compared our model with the baseline models, as shown in Table 1. We used the transformer with base setting [3] as the autoregressive baseline model. Furthermore, the recent strong NAT baseline models were also included for comparison. For all tasks, we obtained the results of other NAT models directly from their original papers if they were available. In addition, we reimplemented the sentence-level semantic agreement [19] proposed for the autoregressive machine translation as SLSAv1.

4.1.4. Model Setup

The hyperparameters used in our model were set closely following the previous works [4,15]. We utilized the small transformer (n_{head} = 4, n_{layer} = 5, and d_{dim} = 256) for the IWSLT task. For the WMT tasks, the base transformer (n_{head} = 8, n_{layer} = 6, and d_{dim} = 512) [3] was used as the model configuration. The weights of our model were randomly initialized using the normal distribution $\mathcal{N}(0, 0.02)$. The Adam optimizer [20] was used to optimize the whole model. The temperature was set as 0.05 and will be analyzed in the next section.

4.1.5. Training and Inference

All our models were trained on 8/1 Nvidia Tesla V100 GPUs with batches of 64 k/8 k tokens for the WMT and IWSLT14 tasks, respectively. We increased the learning rate from 0 to 5 × 10^{-4} during the first 10k steps; then, we decreased the learning rate according to the inverse square root of the training steps [3]. During inference, we averaged the five best checkpoints to create the final checkpoint. The translation was generated using the final checkpoint on one NVIDIA Tesla V100 GPU. Furthermore, we utilized the widely-used BLEU score [21] to evaluate the accuracy of translation.

Table 1. The results of our model on the EN → DE, EN → RO, and DE → EN translation tasks. "NPD" represents noisy parallel decoding. "/" indicates there are no values in this term. "*" denotes the results are obtained by our reimplementation. "k" denotes the number of decoding iterations.

Model			**EN → DE**	**EN → RO**	**DE → EN**
AT Model		**Transformer**	**27.2**	**33.70**	**34.50**
Iterative-Based NAT models		NAT-IR(k = 10) [15]	21.61	29.32	23.94
		LaNAT (k = 4) [22]	26.30	/	/
		LevT (k ≈ 6) [23]	27.27	/	/
		CMLM (k = 4) [24]	25.94	32.53	30.42
		CMLM (k = 10)	27.03	33.08	31.71
		JM-NAT (k = 4) [25]	26.82	32.97	31.27
		JM-NAT (k = 10)	27.31	33.52	32.59
Fully NAT models		NAT [4]	17.69	26.22	/
		Hint-NAT [26]	21.11	/	/
		TCL-NAT [27]	21.94	/	28.16
		DCRF-NAT [6]	23.44	/	/
		Flowseq [28]	21.45	29.34	27.55
		CMLM	18.05	27.32	/
		GLAT [7]	25.21	31.19	29.80 *
		CNAT [29]	25.56	/	31.15
	w/ NPD	NAT (NPD = 100)	19.17	29.79	24.21
		Hint-NAT (NPD = 9)	25.20	/	/
		DCRF-NAT (NPD9)	26.07	/	29.99
		GLAT (NPD = 7)	26.55	32.87	31.23 *
		CNAT (NPD = 9)	26.60	/	/
	Ours	GLAT	25.02	31.09	29.74
		SLSAv1	25.53	31.23	30.45
		SLSA	26.06	31.40	31.14
		SLSA (NPD = 7)	27.01	32.90	32.39

4.2. Main Results

The main results of our model on the three translation tasks are listed in Table 1. Our model achieved a significant improvement and outperformed the other fully NAT models.

Although the performance of the SLSA fell behind the iterative-based models, because it only performs one decoding pass, it has a large speed advantage. In detail, the following observations can be obtained from Table 1:

- Our model, SLSA, is based on the architecture of GLAT, but our model obtained the better results than GLAT. It had a significant improvement (about 0.85 BLEU+) on the EN → DE task. In addition, on the DE → EN task, it gained +1.3 BLEU over GLAT. The reason is that SLSA introduces the semantic agreement between the encoder and decoder. The semantic agreement in the encoder can ensure the encoder projects the representation of the source target sentences into the shared space. The semantic agreement between the source and target sentences can ensure the sentence-level representation of the generated sentence is consistent with the representation of the source sentence. Meanwhile, this semantic agreement gives the decoder a constraint, which limits the decoder to generating semantically relevant words. Furthermore, the result of our model was better than SLSAv1. SLSAv1 uses the mean square error to pull the similar representation closer, but it cannot push the dissimilar representations away. We think this is the reason for better results of the SLSA.
- Compared with the other fully NAT models, our model achieved better results and remained simple. Hint-NAT and TCL-NAT need a pretrained autoregressive model to provide the knowledge for the non-autoregressive model. Although DCRF-NAT does not need a pretrained autoregressive model, it introduces the CRF module and uses viterbi to generate the target sentence, which may reduce the decoding speed when the target sentence is lengthy. However, SLSA only modifies the training process, and it can perform the vanilla decoding process. Compared with CNAT, on the WMT14 EN → DE task, our model achieved a better result.
- As for the NPD, it did have a critical effect on improving the performance of the NAT model, as the previous works [4,25] pointed out. When we reranked the candidates with an autoregressive model, a result of 27.01 BLEU was obtained by our model on the EN → DE task and obtains a comparable result to the autoregressive model transformer, which achieved 27.2 BLEU.

To show the speedup of our model, we present a scatter plot in Figure 2. From Figure 2, we can see that the speedup of the SLSA is consistent with the NAT and GLAT. This is because the decoding process of these models is exactly the same. Compared with the other models, with the same speedup, only the performance of the CMLM and transformer were better than our model. In addition, we can see that reranking with the NPD would significantly increase the decoding latency. Therefore, improving the NAT model without the NPD is worth exploring.

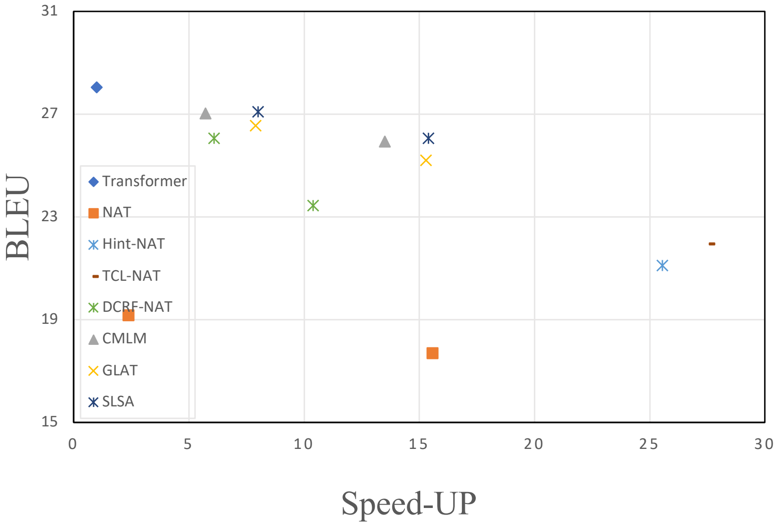


Figure 2. The balance of performance and speedup on the EN → DE.

4.3. Analyses

In this section, the experiments conducted to analyze the performance of SLSA from different aspects are detailed.

4.3.1. Effect of Different Lengths

To evaluate the influence of different target lengths, the different buckets were filled according to the length of the target sentence in the DE → EN test set. The results are shown in Figure 3.

From the results, we can see that as the length increased, the difficulty of translation also increased, and the accuracies of all models decreased. Among them, the accuracy of the vanilla NAT model dropped quickly. In contrast, although the accuracies of GLAT and SLSA also showed a downward trend, they still remained relatively stable. In addition, the SLSA achieved better results under different lengths compared to the GLAT, which demonstrates the effectiveness of our proposed method. What should also be noticed is that when the length was less than 10, the vanilla NAT model achieved better results. We think the reason is that when the length is less than 10, the operation of *glancing* cannot perform well as on a sentence length greater than 10.

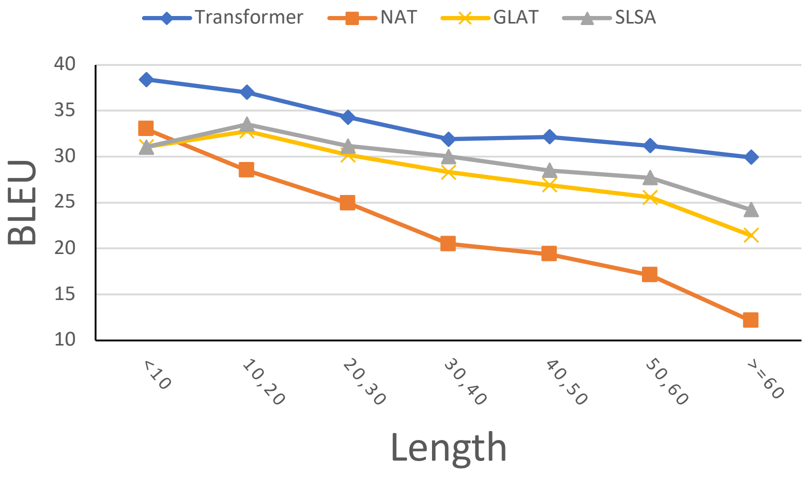


Figure 3. The results from different lengths on the IWSLT14 DE → EN.

4.3.2. Effect on Reducing Word Repetition

A previous work [30] pointed out that word repetition is a critical factor in the performance of the NAT model. We conducted an analysis on the DE → EN validation set to show the effect of our model on reducing word repetition. Following the previous work [30], we counted the number of repetitive tokens per sentence, as shown in Table 2.

Table 2. The average repetitive tokens on the validation set of the DE → EN task.

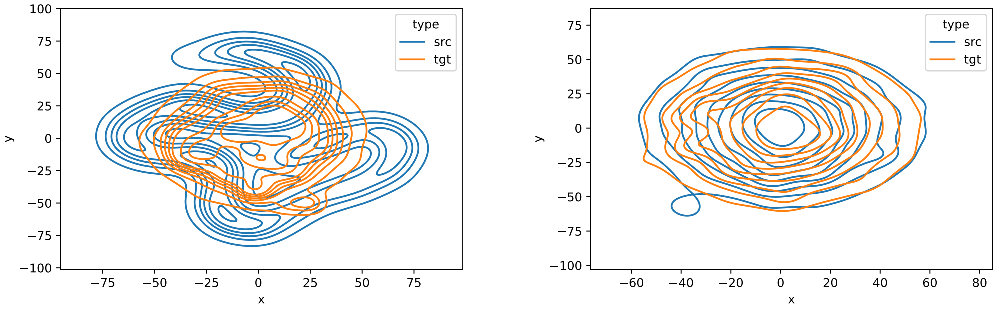
NAT	NAT-Reg	CMLM	GLAT	SLSA
2.30	0.90	0.48	0.49	0.40

The results showed that compared with the vanilla NAT model and NAT-Reg [30], both CMLM and GLAT significantly reduced the number of repeated words, which demonstrated that capturing the target dependency can help to reduce the number of repeated words. CMLM uses an iterative method to capture the target dependency, and GLAT utilizes glancing to capture the target dependency. Compared with GLAT, the semantic agreement in our model further reduced the number of repeated words. This is because more repetition may affect the sentence-level representation of the generated sentence and cause inconsistency between the source and target sentences.

4.3.3. Visualization

In order to more intuitively observe whether our model pulled the source and target representations close together, we visualized the source and target representations. According to Equations (6) and (7), we retrieved the source and target representations of each sentence in the IWSLT14 DE → EN test set.

Following [13], the 256-dim representation was reduced to 2-dim using T-SNE. We then depicted its density estimation, as shown in Figure 4. From Figure 4a, we can see that GLAT did not consider the semantic agreement between the source and target sentences, and the source sentence could not be aligned with the target sentence. In contrast, our model, SLSA, drew the source and target semantic representations much closer together.



(**a**) (**b**)

Figure 4. Bivariate kernel density estimation plots of representations. (**a**) The representations output by GLAT; (**b**) The representations output by SLSA. Source representation is denoted by a blue line, and the target is in orange. This figure shows that our model effectively pulled the source and target sentence-level semantic representations closer together.

4.4. Ablation Study

4.4.1. Influence of Temperature

The temperature τ in the loss $\mathcal{L}_{e2d}$ and $\mathcal{L}_{e2e}$ (Equations (8) and (9)) is used to control the difficulty of distinguishing between the positive and negative samples. A different temperature will result in a different performance. We also conducted an analysis of the influence of temperature on the performance of our model. We used different temperatures from 0.01 to 0.2 to conduct experiments on the DE → EN test set. The results are shown in Figure 5.

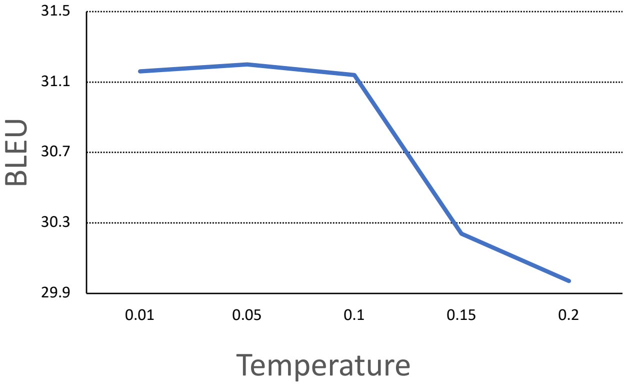


Figure 5. The results from different temperatures on the IWSLT14 DE → EN.

We found that the results varied greatly at different temperatures. When the temperature τ was greater than 0.1, the performance of the SLSA dropped quickly. Furthermore, when the temperature ranged from 0.01 to 0.1, the SLSA achieved similar results. In this work, we set the temperature τ as 0.05 for all experiments.

4.4.2. Effectiveness of Each Loss

To evaluate the contribution of each loss, an analysis was conducted on the DE → EN validation set. The results are listed in Table 3.

Table 3. The effectiveness of each loss on DE → EN validation set.

Model	$\mathcal{L}_{A_{GT}}$	$\mathcal{L}_{e2e}$	$\mathcal{L}_{e2d}$	BLEU
SLSA				30.34
	✓			30.51
	✓	✓		30.98
	✓		✓	30.82
	✓	✓	✓	31.19

$\mathcal{L}_{A_{GT}}$ was the base of our method, so we first trained our model only with this loss. When adding this loss to the model, a better result was achieved. Then, we added the loss $\mathcal{L}_{e2e}$ and $\mathcal{L}_{e2d}$ to the model separately. Both $\mathcal{L}_{e2e}$ and $\mathcal{L}_{e2d}$ improved the performance of the SLSA. When adding all the losses to the model, the performance of the SLSA was further improved.

5. Related Work

5.1. Non-Autoregressive Machine Translation

Since the non-autoregressive translation was proposed [4], several methods have been introduced to improve the performance of the NAT model. From modeling the input by latent variables [4,22,28] in the early stage to capturing the target dependency [7,15,24,25], the performance of the NAT model has been greatly improved. In addition, there are some works that have used different training objectives to improve the performance of the NAT model [5,6,30,31]. However, there is no work considering the semantic consistency between the source and target sentences. In addition, our work is related to the DLSP [32]. The DSLP adds a supervised signal at each layer of the decoder, but we only added a supervised signal at the penultimate layer. In addition, we used an ahead supervised signal to obtain better representations, which was different from the DSLP.

5.2. Sentence-Level Agreement

In machine translation, the semantics of the source and target sentences should be consistent. For autoregressive translation, there has been some work exploring the effectiveness of sentence-level agreement. Yang et al. [33] utilized the mean square error to pull the representations of source and target sentences closer together. Furthermore, Yang et al. [19] introduced a new method, which extracted the representations of the source and target sentences and pulled the representations closer layer by layer with the MSE error. However, they only considered the semantic agreement between the source and target sentences for the autoregressive model. In contrast, we attempted to use contrastive learning to pull the representations closer together for the non-autoregressive translation. In addition, we considered not only the semantic agreement between the source and target sentences but also that the semantic representation output by the encoder remained consistent with the target representation.

5.3. Contrastive Learning

Contrastive learning has achieved great success in various computer vision tasks [34–37]. Given the performance of contrastive learning, researchers in NLP have also attempted to use it for sentence representation [10,11,38,39]. Compared to the success in sentence representation, it is difficult to apply contrastive learning on machine translation. Recently, Pan et al. [13] utilized contrastive learning to project multilingual sentence representations into a shared space, which improved the performance of multilingual machine translation. Inspired by this, we utilized contrastive learning to ensure the sentence level semantic representations of the source and target sentences remained consistent.

6. Conclusions

In this work, we utilized contrastive learning to obtain the sentence-level semantic agreement between the source and target sentences. In addition, to strengthen the capability of the encoder, we also added an agreement module to the encoder to project the source and target sentences into a shared space. Experiments and analyses were conducted on three translation benchmarks, and the results showed that our model improved the performance of the NAT model. In the future, we will consider applying contrastive learning to improving the word-level or phrase-level semantic agreement. In addition, our method selected negative samples from the same batch, which may have similar semantics. So, in the future, we will explore more methods from which to select negative samples.

Author Contributions: Project administration, S.W.; supervision, H.H. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 61732005 and 61671064.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Our work was supported by the National Natural Science Foundation of China (Nos. 61732005, 61671064).

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

References

1. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
2. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional Sequence to Sequence Learning. In Proceedings of the ICML 2017, Sydney, Australia, 6–11 August 2017 .
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017 ; pp. 5998–6008.
4. Gu, J.; Bradbury, J.; Xiong, C.; Li, V.O.; Socher, R. Non-autoregressive neural machine translation. *arXiv* **2017**, arXiv:1711.02281.
5. Shao, C.; Zhang, J.; Feng, Y.; Meng, F.; Zhou, J. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 198–205.
6. Sun, Z.; Li, Z.; Wang, H.; Lin, Z.; He, D.; Deng, Z.H. Fast structured decoding for sequence models. *arXiv* **2019**, arXiv:1910.11555.
7. Qian, L.; Zhou, H.; Bao, Y.; Wang, M.; Qiu, L.; Zhang, W.; Yu, Y.; Li, L. Glancing transformer for non-autoregressive neural machine translation. *arXiv* **2020**, arXiv:2008.07905.
8. Saharia, C.; Chan, W.; Saxena, S.; Norouzi, M. Non-autoregressive machine translation with latent alignments. *arXiv* **2020**, arXiv:2004.07437.
9. 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, Virtual, 13–18 July 2020; pp. 3515–3523.
10. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv* **2021**, arXiv:2104.08821.
11. Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; Xu, W. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *arXiv* **2021**, arXiv:2105.11741.
12. Wang, R.; Finch, A.; Utiyama, M.; Sumita, E. Sentence embedding for neural machine translation domain adaptation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 560–566.
13. Pan, X.; Wang, M.; Wu, L.; Li, L. Contrastive learning for many-to-many multilingual neural machine translation. *arXiv* **2021**, arXiv:2105.09501.

14. Liu, Y.; Wan, Y.; Zhang, J.; Zhao, W.; Yu, P. Enriching non-autoregressive transformer with syntactic and semantic structures for neural machine translation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Kiev, Ukraine, 21–23 April 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 1235–1244. [CrossRef]
15. Lee, J.; Mansimov, E.; Cho, K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1173–1182.
16. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1715–1725.
17. Zhou, C.; Neubig, G.; Gu, J. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv* **2019**, arXiv:1911.02727.
18. Kim, Y.; Rush, A.M. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1317–1327.
19. Yang, M.; Wang, R.; Chen, K.; Wang, X.; Zhao, T.; Zhang, M. A Novel Sentence-Level Agreement Architecture for Neural Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2585–2597. [CrossRef]
20. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
21. 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, Philadelphia, PA, USA, 7–12 June 2002; pp. 311–318.
22. 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 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8846–8853.
23. Gu, J.; Wang, C.; Zhao, J. Levenshtein transformer. *arXiv* **2019**, arXiv:1905.11006.
24. Ghazvininejad, M.; Levy, O.; Liu, Y.; Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. *arXiv* **2019**, arXiv:1904.09324.
25. Guo, J.; Xu, L.; Chen, E. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual, 5–10 July 2020; pp. 376–385.
26. Li, Z.; He, D.; Tian, F.; Qin, T.; Wang, L.; Liu, T.Y. Hint-based training for non-autoregressive translation. In Proceedings of the ICLR 2019 Conference, New Orleans, LA, USA, 6–9 May 2018.
27. Guo, J.; Tan, X.; Xu, L.; Qin, T.; Chen, E.; Liu, T.Y. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7839–7846.
28. Ma, X.; Zhou, C.; Li, X.; Neubig, G.; Hovy, E. Flowseq: Non-autoregressive conditional sequence generation with generative flow. *arXiv* **2019**, arXiv:1909.02480.
29. Bao, Y.; Huang, S.; Xiao, T.; Wang, D.; Dai, X.; Chen, J. Non-Autoregressive Translation by Learning Target Categorical Codes. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Virtual, 6–11 June 2021; pp. 5749–5759.
30. Wang, Y.; Tian, F.; He, D.; Qin, T.; Zhai, C.; Liu, T.Y. Non-autoregressive machine translation with auxiliary regularization. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5377–5384.
31. Shao, C.; Feng, Y.; Zhang, J.; Meng, F.; Chen, X.; Zhou, J. Retrieving sequential information for non-autoregressive neural machine translation. *arXiv* **2019**, arXiv:1906.09444.
32. Huang, C.; Zhou, H.; Zaiane, O.R.; Mou, L.; Li, L. Non-autoregressive translation with layer-wise prediction and deep supervision. *arXiv* **2021**, arXiv:2110.07515.
33. Yang, M.; Wang, R.; Chen, K.; Utiyama, M.; Sumita, E.; Zhang, M.; Zhao, T. Sentence-level agreement for neural machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3076–3082.
34. Zhuang, C.; Zhai, A.L.; Yamins, D. Local aggregation for unsupervised learning of visual embeddings. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6002–6012.
35. Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 776–794.
36. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 4116–4126.
37. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9729–9738.
38. Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. Clear: Contrastive learning for sentence representation. *arXiv* **2020**, arXiv:2012.15466.
39. Fang, H.; Wang, S.; Zhou, M.; Ding, J.; Xie, P. Cert: Contrastive self-supervised learning for language understanding. *arXiv* **2020**, arXiv:2005.12766.