SemFace: Pre-training Encoder and Decoder with a Semantic Interface for Neural Machine Translation

Shuo Ren[†‡*], **Long Zhou**[‡], **Shujie Liu**[‡], **Furu Wei**[‡], **Ming Zhou**[‡], **Shuai Ma**[†]
[†]SKLSDE Lab, Beihang University, Beijing, China
[‡]Microsoft Research Asia, Beijing, China
[†]{shuoren,mashuai}@buaa.edu.cn [§]{long.zhou,shujliu,fuwei,mingzhou}@microsoft.com

Abstract

While pre-training techniques are working very well in natural language processing, how to pre-train a decoder and effectively leverage it for neural machine translation (NMT) still remains a tricky issue. The main reason is that the cross-attention module between the encoder and decoder cannot be pre-trained, and the combined encoder-decoder model cannot work well in the fine-tuning stage because the inputs of the decoder cross-attention come from unknown encoder outputs. In this paper, we propose a better pre-training method for NMT by defining a semantic interface (**SemFace**) between the pre-trained encoder and the pre-trained decoder. Specifically, we propose two types of semantic interfaces, including **CL-SemFace** which regards cross-lingual embeddings as an interface, and **VQ-SemFace** which employs vector quantized embeddings to constrain the encoder outputs and decoder inputs in the same language-independent space. We conduct massive experiments on six supervised translation pairs and three unsupervised pairs. Experimental results demonstrate that our proposed SemFace can effectively connect the pre-trained encoder and decoder, and achieves significant improvement by 3.7 and 1.5 BLEU points on the two tasks respectively compared with previous pre-training-based NMT models.

1 Introduction

In recent years, pre-trained language models (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019; Raffel et al., 2020) significantly boost the performances of various natural language processing (NLP) tasks, receiving extensive attention in NLP communities. Following the idea of unsupervised pre-training methods in the NLP area, several approaches (Lample and Conneau, 2019; Zhu et al., 2020; Lewis et al., 2020; Liu et al., 2020) have been proposed to improve neural machine translation (NMT) models with pre-training by leveraging the large-scale monolingual corpora. The typical training process usually consists of two stages: pre-training an encoder and a decoder separately with a large monolingual corpus in a self-supervised manner, and then fine-tuning on specific NMT tasks (Lample and Conneau, 2019).

The above method essentially pre-trains a BERT-like (Devlin et al., 2019) Transformer encoder, and uses it to initialize both the encoder and decoder. Although it shows promising results, pre-training decoder benefits little in their results. The potential reason is that the cross-attention between the encoder and decoder is not pre-trained, which is randomly initialized when they are connected for fine-tuning, resulting in a lack of semantic interfaces between the pre-trained encoder and decoder. Another line of work attempts to pre-train a sequence-to-sequence model directly, e.g., MASS (Song et al., 2019) and BART (Lewis et al., 2020). But these methods usually use monolingual denoising auto-encoder as the main training objective, and cannot learn the corss-lingual mapping between source and target languages explicitly.

In parallel to the idea of DALL·E[1] which defines the cross-modality interface of image and text, we propose to pre-train the encoder and decoder with a language-independent semantic interface (**SemFace**) for neural machine translation. With the semantic interface, the encoder is pre-trained to extract features to this space, and the decoder is pre-trained to generate contents with features provided by it. By defining this interface, we can decouple the encoder-decoder network and pre-train them separately. During the decoder pre-training, the cross-attention module is also pre-trained, thus the pre-trained encoder and decoder can be naturally

[*]Contribution during internship at MSRA.

[1]https://openai.com/blog/dall-e/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4518–4527
August 1–6, 2021.

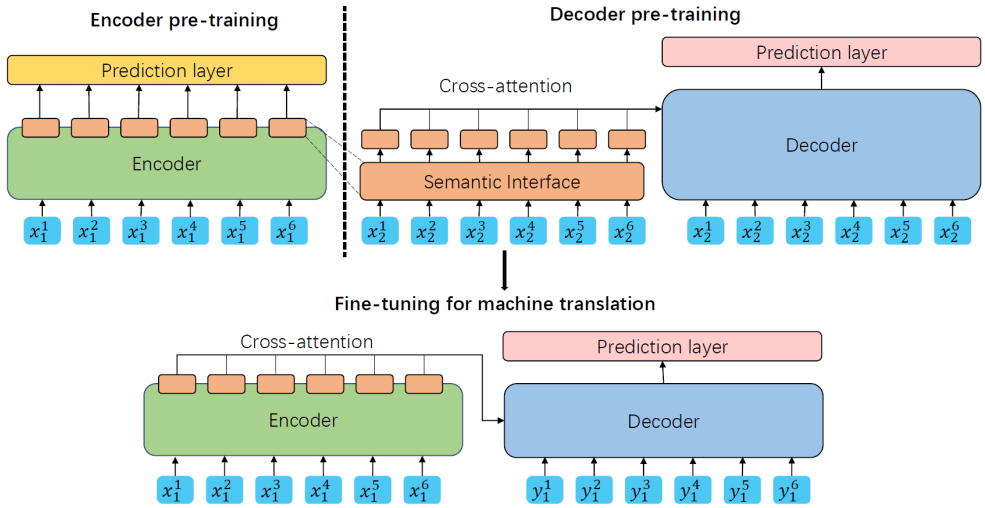

Figure 1: Overview of our method (Top: pre-training; Bottom: fine-tuning). The training steps of pre-training encoder and decoder are separated, therefore the training samples of them are not necesarrily the same. (In the figure, the training sample for pre-training the encoder is $\mathbf{x_1} = x_1^1 x_1^2 .. x_1^6$) and the training sample for pre-training the decoder is $\mathbf{x_2} = x_2^1 x_2^2 .. x_2^6$). For MT fine-tuning, we use the parallel training sample $\{\mathbf{x_1}, \mathbf{y_1}\}$ from the parallel corpus or generated from back-translation.

connected for MT fine-tuning. We propose two types of semantic interfaces, namely **CL-SemFace** and **VQ-SemFace**. The former takes the trained unsupervised cross-lingual embeddings (Artetxe et al., 2018) as the interface for encoder and decoder pre-training. Inspired by the success of neural discrete representation learning (Van Den Oord et al., 2017), the latter uses language-independent vector quantized (VQ) embeddings (semantic unites) as the interface to map encoder outputs and decoder inputs into the shared VQ space. Experiments conducted on both supervised and unsupervised translation tasks demonstrate that SemFace effectively connects the pre-trained encoder and decoder, and achieves a significant improvement by 3.7 and 1.5 BLEU points on the two tasks respectively.

Our contributions are listed as follows:

- To the best of our knowledge, this is the first work to investigate and define a semantic interface between encoder and decoder for the MT pre-train-finetune framework.
- We design and compare two effective types of semantic interfaces, which utilize cross-lingual embeddings and vector quantized embeddings respectively.
- We extensively verify the effectiveness of our proposed model on supervised and unsupervised NMT tasks. Particularly, our proposed CL-SemFace and VQ-SemFace lead to significant improvements of 3.38 and 3.76 BLUE points on low-resource language pairs.

2 SemFace

2.1 Pre-training both Encoder and Decoder

The overview of our proposed SemFace is illustrated in Figure 1. As shown in this figure, our method can be divided into two steps. First, we use monolingual data to pre-train encoder and decoder separately with a semantic interface between them. The encoder is pre-trained to map the input from the monolingual semantic space into the interface, while the decoder is pre-trained to use the content from the interface via the cross attention module to finish decoding. The parameters of the encoder and the decoder are updated independently, thus their pre-training processes can be either jointly or separately done. Then, we remove the semantic interface, and connect the pre-trained encoder and decoder with the pre-trained cross-attention as a sequence-to-sequence model for the subsequent machine translation fine-tuning. Note that in Figure 1, the input to the encoder and decoder includes token representations, language embeddings and positional embeddings.

There are three types of semantic interface. The first is the default output space of pre-trained encoder with the masked language model (MLM) training loss. In fact, previous work (Song et al., 2019; Lewis et al., 2020; Liu et al., 2020) adopts this default settings in their pre-training method for machine translation. The second one is **CL-**

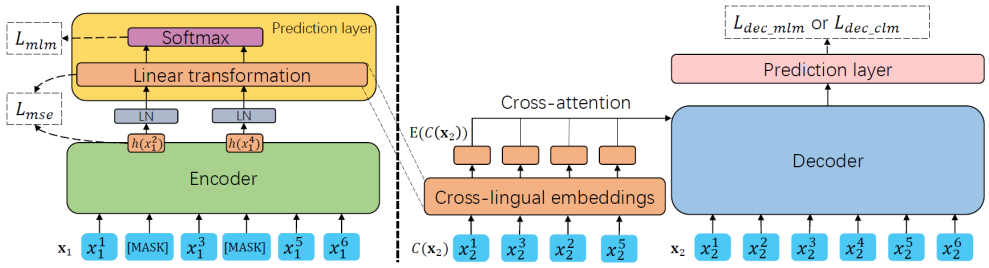

Figure 2: CL-SemFace, which regards a pre-trained cross-lingual embeddings as a semantic interface.

Algorithm 1: Pre-training with SemFace

Input: Monolingual corpora D_X and D_Y for two languages
Output: the MT model M_θ

Randomly initialize the parameters of the encoder θ_{enc} and the decoder θ_{dec} as well as the semantic interface θ_{sf}
Initialize θ_{sf} with pre-trained cross-lingual embeddings (for **CL-SemFace**)
while *not convergence* **do**
  Sample a batch B from D_X or D_Y
  Pass B through the encoder with SemFace
  Update θ_{enc} and θ_{sf}
  Pass B through the decoder with SemFace
  Update θ_{dec}
return $M_\theta = \{\theta_{enc}, \theta_{dec}\}$

SemFace (Sec. 2.2), which uses the pre-trained context-free cross-lingual embedding space as the semantic interface. The third is **VQ-SemFace** (Sec. 2.3), which automatically learns a context-aware vector quantized (VQ) embedding space as the interface during pre-training. The last two types define a language-independent interface, enforcing the pre-trained encoder and the decoder to generate or leverage the language-independent information. They can provide a better initialization for the following MT fine-tuning. We give our pre-training algorithm in Alg. 1. Note that the parameters of the cross-attention are included in θ_{dec}. Next, we will introduce our proposed CL-SemFace and VQ-SemFace in detail.

2.2 CL-SemFace

CL-SemFace uses the cross-lingual embedding space as the interface between the encoder and the decoder during pre-training. We first concatenate the monolingual corpora of two languages and learn joint BPE, and then train cross-lingual BPE embeddings with VecMap (Artetxe et al., 2018).

As shown in Figure 2, on the encoder side, we initialize the linear projection weights (output embeddings) before the Softmax with the pre-trained BPE embeddings, and pre-train the encoder with two training objectives. The first is the commonly used Masked Language Model (MLM) (Devlin et al., 2018) l_{mlm}, and the other is the MSE loss l_{mse} between the encoder output hiddens and the corresponding output embeddings. The latter controls the scale of the encoder outputs to be the same as the cross-lingual embeddings, in order to match the encoder outputs and the cross-attention inputs. To stabilize training, we calculate the MSE loss before the last normalization layer of the encoder. Formally, given an input sample $\mathbf{x}$, the encoder pre-training loss function is:

$$\begin{aligned}\mathcal{L}_{enc} =& \mathcal{L}_{mlm} + \mathcal{L}_{mse} \\ =& \sum_i [-\log p(x_i|\mathrm{LN}(h_i(\mathbf{x}))) \\ &+ (\mathbf{W}_i - h_i(\mathbf{x}))^2]\end{aligned} \quad (1)$$

where x_i is the masked tokens in the input sentence, h_i is the activation of the final layer of the encoder but before the final layer normalization LN, W_i is the output embedding of the ground-truth token, and p is the output probability of the Softmax.

When pre-training the decoder, we attempt to use the content from the semantic interface to simulate encoder outputs. To achieve that, given a monolingual training sample $\mathbf{x}$, we first add some noise[1] into it to get the noisy sample $C(\mathbf{x})$), then we pass it through an embedding layer initialized with the pre-trained BPE embeddings to get the language-independent representations $E(C(\mathbf{x}))$. The training target of the decoder is either the MLM or the Casual Language Model (CLM) (Lample and Conneau, 2019). Different from them, in our work, the decoder is trained to generate contents with the language-independent representations from the semantic interface. During this process, the parameters of the enc-dec attention (cross-attention) can also be pre-trained, which is critical to the subsequent machine translation fine-tuning. Formally,

[1]The noise here includes words dropping and swapping as in Lample et al. (2018).

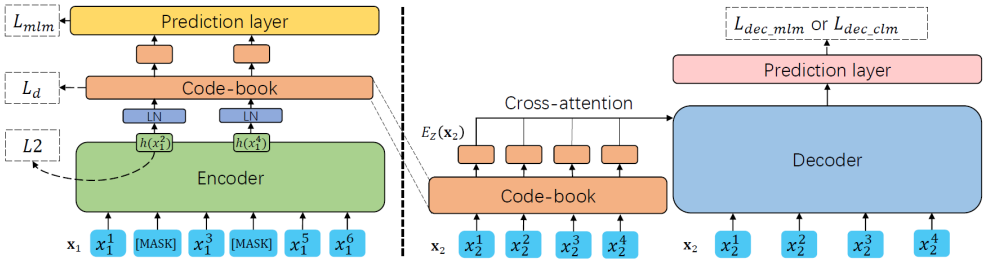

Figure 3: VQ-SemFace, which utilizes vector quantized embeddings as a semantic interface.

the decoder pre-training loss functions is:

$$\mathcal{L}_{dec\_mlm} = \sum_{j} -\log p[y_j|(s_j(\mathbf{x})), E(C(\mathbf{x}))] \tag{2}$$

or

$$\mathcal{L}_{dec\_clm} = \sum_{j} -\log p(y_j|(s_{<j}(\mathbf{x})), E(C(\mathbf{x}))] \tag{3}$$

where s is the final output hidden of the decoder and p is the output probability of the Softmax.

2.3 VQ-SemFace

The CL semantic space is constrained with the cross-lingual word embedding, which is context-independent, meaning that the different meanings of the same word share the same embedding, and the number of semantic units should be the same with the size of the vocabulary. In order to learn context-dependent semantic units freely, we also propose another interface type, vector quantized embeddings, inspired by the recent success of VQ-based speech pre-training (Baevski et al., 2020). The concept of Vector Quantized (VQ) representations is first proposed in Van Den Oord et al. (2017). The method uses a learnable code-book combined with the nearest neighbor search to train the discrete latent variable model. The code-book is essentially a group of learnable embeddings (codes) $\{z\}_1^K$. The nearest neighbor search is performed between the encoder outputs and the embedding of the latent code using the $\mathcal{L}2$ distance metric. Formally, given the encoder output $h(\mathbf{x})$, the discrete latent variable assignment is given by

$$z_i = \arg\min_{j \in [K]} ||h(\mathbf{x}) - z_j||_2 \tag{4}$$

where K is the number of codes in the code-book, z_j is j-th quantized vector in the code-book. That means, z_i is the output of the VQ layer corresponding to $h(\mathbf{x})$. The main issue of this method is that the $\arg\min$ operation is not differentiable. Following Baevski et al. (2020), we use the Gumbel-Softmax (Gumbel, 1954; Jang et al., 2016) to select discrete codebook variables in a fully differentiable way and we use the straight-through estimator of Jang et al. (2016). Given the encoder output $h(\mathbf{x})$, we apply a linear layer followed by a ReLU and another linear which outputs $l \in \mathbb{R}^K$ logits for the Gumbel-Softmax. During inference, we simply pick the largest index in l. During training, the output probability to choose the j-th code is

$$p_j = \frac{\exp(l_j + v_j)/\tau}{\sum_{k=1}^{K} \exp(l_k + v_k)/\tau} \tag{5}$$

where $v = -\log(-\log(u))$ and u are uniform samples from $\mathcal{U}(0, 1)$. In the forward pass, only the embedding in the code-book with the largest probability is used, which means the output of the VQ layer is z_i, where $i = \arg\max_i p_i$, while in the backward pass, the gradient is passed to all the Gumbel-Softmax outputs.

The VQ layer groups the context-aware hidden states into limited semantic units (codes), and the space of these codes can be used as our second language-independent semantic interface. As shown in Figure 3, for the encoder, we add a VQ layer between the encoder output and the prediction layer of MLM. The training loss is the combination of the original MLM loss and two auxiliary losses as used in Baevski et al. (2020). The first is the diversity loss $\mathcal{L}_d$ to encourage the model to use the code-book entries equally often by maximizing the entropy of the averaged Softmax distribution over the codes across a batch of utterances as

$$\mathcal{L}_d = \frac{1}{K} \sum_{k=1}^{K} \bar{p}_k \log \bar{p}_k \tag{6}$$

where $\bar{p}_k$ is the averaged probability of choosing the k-th code in the code-book across a batch, and p_k is calculated by Eq.(5). The second auxiliary loss is an $\mathcal{L}2$ penalty to stabilize the training, which

is applied to the activations of the final encode layer but before the last normalization of the encoder. Therefore, the total loss of encoder pre-training is $\mathcal{L}_{enc} = \mathcal{L}_{mlm} + \mathcal{L}_d + \mathcal{L}2$.

For the decoder, similar to CL-SemFace, we also use the content from the VQ interface to simulate the encoder output during pre-training. To get the VQ output, given a training sample, we first feed it into an embedding layer and then pass the readout embeddings to a two-layer Transformer, which can be regarded as a feature extractor. We use the Transformer output as the representations of each word and find the corresponding codes in the code-book according to Eq.(5). The readout codes are the simulated encoder output, and they will be fed into the decoder via the cross-attention. Note that in the decoder pre-training, the VQ code-book is fixed. The training goal of the decoder is the same as that in CL-SemFace, i.e., $\mathcal{L}_{dec\_mlm}$ or $\mathcal{L}_{dec\_clm}$.

2.4 Fine-tuning

The semantic interface acts as a bridge to connect the encoder and decoder during pre-training. The encoder is pre-trained to project the input to the features in the semantic interface space, while the decoder is pre-trained to leverage the features from the interface space through the cross-attention to generate outputs. With this method, we can pre-train all the parameters of the whole sequence-to-sequence model, including the cross-attention between the encoder and the decoder. After pre-training, we connect the encoder and the decoder via the cross-attention directly by removing the semantic interface as shown in Figure 1 (bottom). We then fine-tune the model on low-resource supervised NMT tasks and unsupervised NMT tasks. For the low-resource settings, we use the standard cross-entropy loss $-\log p(\mathbf{y}|\mathbf{x})$ given the parallel training sample $\{\mathbf{x}, \mathbf{y}\}$, and for the unsupervised settings, we use the denoising auto-encoder and iterative back-translation as the objectives as in Lample and Conneau (2019).

3 Experiment

3.1 Setup

3.1.1 Dataset

The languages we choose for our experiments are English (en), French (fr), German (de), Romanian (ro), Finnish (fi), Estonian (et), Latvian (lv), Lithuanian (lt), Gujarati (gu), and Kazakh (kk). The details of the datasets and statistics for each language pair are listed in Table 1. All the data is provided by the recent WMT translation tasks. "Para Data" in this table means the number of training samples of "x-en". The language pairs with parallel data in the table are chosen for the low-resource supervised settings, while those with only monolingual data are chosen for the unsupervised scenario only. For the language with more than 50 million monolingual data, we randomly sample 50 million from the corpus. We choose the corresponding development and test sets for each language pair from WMT translation tasks, as listed in Table 2.

Lang	Mono Data Source	#Sent	Para Data
en	NC	50M	-
fr	NC	50M	-
de	NC	50M	-
ro	NC	21M	-
fi	NC, CC	50M	2.7M
et	NC, CC, BE	50M	1.9M
lv	NC, CC	38M	4.5M
lt	NC, CC, Wiki	50M	2.1M
gu	NC, CC, Wiki	4.3M	10K
kk	NC, CC, Wiki	12.7M	91K

Table 1: The datasets used in our experiments. Lang: language; Mono: monolingual; Para: parallel; #Sent: number of sentences in the monolingual corpus; NC: NewsCrawl; CC: CommonCrawl; BE: BigEst Estonian corpus; Wiki: Wiki dumps.

Language-pair	Dev set	Test set
en-fr	*newstest*2013	*newstest*2014
en-de	*newstest*2013	*newstest*2016
en-ro	*newsdev*2016	*newstest*2016
en-fi	*newsdev*2015	*newstest*2017
en-et	*newsdev*2018	*newstest*2018
en-lv	*newsdev*2017	*newstest*2017
en-lt	*newsdev*2019	*newstest*2019
en-gu	*newsdev*2019	*newstest*2019
en-kk	*newsdev*2019	*newstest*2019

Table 2: Development and test sets for each pair.

3.1.2 Baselines

We compare our method with two baselines. The first is XLM (Lample and Conneau, 2019), which pre-trains a Transformer encoder with the MLM or CLM loss and then initializes the encoder and the decoder with the pre-trained model. The parameters of the cross-attention module are randomly initialized. The second baseline is mBART (Liu et al., 2020), which pre-trains the whole sequence-to-sequence architecture with the denoising auto-encoder loss on the multilingual corpus. For a fair

Method	en-fi		en-et		en-lt		en-lv		en-gu		en-kk		avg.
	→	←	→	←	→	←	→	←	→	←	→	←	
Transformer	20.3	21.7	17.7	22.4	12.2	18.1	12.7	15.4	0.0	0.1	0.2	0.8	11.80
XLM	21.1	25.4	20.6	24.9	14.5	20.7	14.2	17.8	0.0	0.0	1.7	4.5	(+1.98)
mBART	21.9	**26.7**	20.8	25.8	14.7	20.4	14.6	18.7	0.1	0.3	2.1	6.3	(+2.57)
CL-SemFace	**22.7**	25.1	**21.8**	26.6	**15.9**	21.8	**15.9**	19.7	0.5	1.9	2.7	7.6	(+3.38)
VQ-SemFace	22.1	25.3	21.6	**27.0**	15.4	**22.3**	15.4	**20.1**	**1.7**	**2.6**	**3.8**	**9.4**	(+**3.76**)

Table 3: BLEU scores of the low-resource language pairs. Baseline results are based on our reproduction. The last row means the averaged improvement of each method compared with the basic Transformer without pre-training.

comparison, we use their pre-training method on the concatenated corpora of each language pair, i.e., mBART02 in their paper. For the low-resource supervised settings, we also compare our method with the basic Transformer without pre-training. If there is a parallel corpus for a certain language pair, we use the parallel data to fine-tune the pre-trained models in the two baselines. If there is only a monolingual corpus, we use the denoising auto-encoder and iterative back-translation to fine-tune the pre-trained models.

3.1.3 Implementation Details

We implement our method based on the code released by Lample and Conneau (2019). For each language pair, we first lower-case all the case-sensitive languages by default and pre-process the concatenated corpora of each language pair with 60,000 joint BPE codes. For both encoder and decoder, we use 6-layer Transformers with the embedding and hidden dimensions of 1024, 8 attention heads, and a dropout rate of 0.1. The maximum sequence length is 256 and the batch size is 128. We use the Adam optimizer (Kingma and Ba, 2014) for both pre-training and fine-tuning. During pre-training, the learning rate is 0.0001 constantly. During MT fine-tuning, the learning rate is 0.0001 with 4,000 warm-up steps, and then decayed based on the inverse square root of the update number. The loss of the denoising auto-encoder objective is weighted by a coefficient α, and it is linearly decreased to 0.1 in the first 100,000 steps and decreased to 0 in the next 200,000 steps. For VQ-SemFace, the code-book contains 102,400 codes with their dimensions being 1024.

3.2 Main Results

In this section, we report the result of our pre-training method fine-tuned with neural machine translation. We have two settings. The first setting is low-resource supervised machine translation, which uses additional parallel corpus to fine-tune the pre-trained encoder and decoder. The second is unsupervised neural machine translation, which uses the two objectives of denoising auto-encoder and back-translation to fine-tune the model.

3.2.1 Low-resource Language Pairs

The results on the low-resource language pairs are shown in Table 3. From the table, we see that our proposed methods CL-SemFace and VQ-SemFace significantly outperform the non-pre-training Transformer with an average improvement of over 3 BLEU scores. Compared with the strong baseline mBART, our methods also outperform it by 0.8 to 1.2 BLEU scores. For most translation directions, VQ-SemFace is better than CL-SemFace, maybe due to the lower quality of cross-lingual language embeddings of these language pairs, especially for the distant language pairs (en-gu and en-kk). This also shows the shortcomings of the CL-SemFace that it depends on the quality of the cross-lingual embeddings. If the quality is not good, the semantic interface will be far from language-independent, posing difficulties for the splicing of the pre-trained encoder and the pre-trained decoder. By contrast, VQ-SemFace gets rid of the constraints of cross-lingual embeddings and learns a context-dependent semantic space shared across languages, which can handle those language pairs with low-quality cross-lingual embeddings better.

3.2.2 Unsupervised Language Pairs

We also report the results of three unsupervised language pairs in Table 4. From the table, we find our proposed methods also significantly outperform the baseline XLM over 1 BLEU score. Compared with mBART, we also obtain an improvement of nearly 0.9 BLEU score (CL-SemFace). Contrary to the result of low-resource pairs in Table 3, for the language pairs in Table 4, we see the performance of CL-SemFace is better than VQ-SemFace. This

Method	en-fr		en-de		en-ro		avg.
	→	←	→	←	→	←	
XLM	33.0	33.4	26.4	34.3	33.1	31.5	31.95
mBART	33.1	32.9	**29.8**	34.0	33.7	30.9	(+0.45)
CL-SemFace	**34.3**	**35.0**	28.8	**35.2**	**34.5**	**32.9**	(+**1.50**)
VQ-SemFace	34.2	34.5	28.6	34.8	33.9	32.5	(+1.13)

Table 4: BLEU scores of three unsupervised language pairs. Baseline results are based on our reproduction. The last row means the averaged improvement of each method compared with the XLM.

may be because the cross-lingual embeddings of these rich-resource language pairs are of higher quality, thus the semantic interface is initialized better during the pre-training.

3.3 Discussion

3.3.1 Ablation Study

In this subsection, we first investigate the influence of the encoder losses (Eq. 1) by removing each of them independently in the encoder pre-training. Besides, note that there are two types of loss used in our decoder pre-training, MLM and CLM, as shown in Eq. (2,3), so we also compare the results with different losses in decoder pre-training, taking the supervised pair en-fi and unsupervised pair en-ro as examples.

Method	en-fi		en-ro		avg.
	→	←	→	←	
Encoder Pre-training Loss					
CL-SemFace	22.7	25.1	34.5	32.9	28.80
-$\mathcal{L}_{mse}$	21.3	24.6	33.3	31.6	(-1.10)
VQ-SemFace	22.1	25.3	33.9	32.5	28.45
-$\mathcal{L}_d$	19.7	17.4	29.8	29.6	(-4.33)
-$\mathcal{L}2$	21.4	24.5	32.5	31.5	(-0.97)
Decoder Pre-training Loss					
CL-SemFace (MLM)	22.4	**25.1**	**34.5**	**32.9**	**28.73**
CL-SemFace (CLM)	**22.7**	24.7	33.9	32.1	28.35
VQ-SemFace (MLM)	**22.1**	25.1	**33.9**	**32.5**	**28.40**
VQ-SemFace (CLM)	21.9	**25.3**	33.2	31.9	28.08

Table 5: Ablation study of each loss in pre-training.

From the table, we find that for VQ-SemFace under encoder pre-training, the most influential auxiliary loss is the diversity loss $\mathcal{L}_d$, which contributes 4.33 BLEU scores in the final results, which is designed to encourage the model to use the code-book entries equally often. According to our observation, without $\mathcal{L}_d$, the model only uses a small group of codes in the code-book ($< 30\%$), which indeed shrinks the VQ semantic space and leads to the bad performance. $\mathcal{L}_{mse}$ and $\mathcal{L}2$ have a similar effect that stabilizes the training, contributing about 1 BLEU score in the final result. For decoder pre-training, the performance of the two losses is comparable, with the MLM slightly better.

3.3.2 Influence of Parallel Data

In this section, we investigate the influence of the data quantity in the experiments. The language pair we choose is de-en, which has a large parallel corpus and makes it possible to conduct our investigation. We compare the performance of the model with our pre-training method and the model without pre-training. Note that we do not use any monolingual data in the training so the result here is not comparable with that in Table 4.

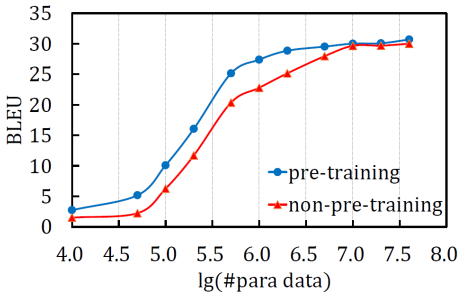

Figure 4: Test BLEU of de-en wt./wto. pre-training. The horizontal axis is $\log_{10}$ of the used parallel data.

As shown in Figure 4, when the number of parallel training data is less than $10^{6.7} \approx 5$M, the model with pre-training significantly outperforms the non-pre-training model by about 3 to 5 BLEU scores. However, when the training samples increase to over 10M, there is almost no difference in performance between the two models.

3.3.3 Analysis about VQ

As mentioned in Sec.2.3, VQ space could be regarded as a language-independent semantic interface for the encoder and decoder pre-training. To test whether VQ space is trained to contain cross-lingual representations, we carry out an analysis with a parallel sample of de-en. For each token pair

(w_{en}, w_{de}) in the two sentences, we collect top-100 codes according to Eq. (5), and calculate how much the codes overlapped, as $\frac{\text{code}_{100}(w_{en}) \cap \text{code}_{100}(w_{de})}{100}$. As shown in Figure 5, the translated tokens share much of the codes chosen from the VQ code-book, which verifies our motivation that VQ could act like a language-independent semantic interface.

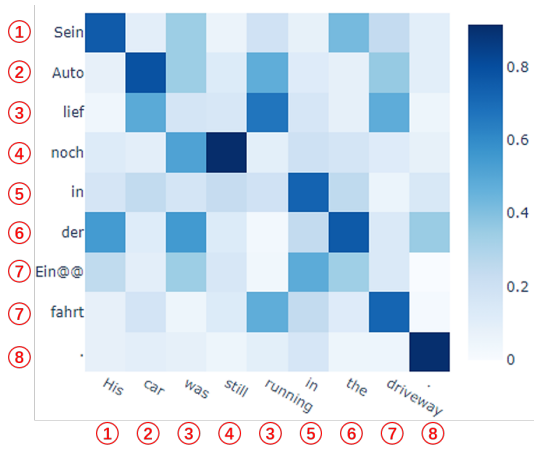

Figure 5: The percentage of the overlapping codes chosen for each token pair. The red numbers denote the translated tokens.

4 Related Work

Pre-training has been widely used in NLP tasks to learn better language representations (Peters et al., 2018; Devlin et al., 2018; Lample and Conneau, 2019; Radford et al., 2019; Yang et al., 2019; Dong et al., 2019; Lewis et al., 2020). Typically, these methods first pre-train neural networks on large-scale unlabeled corpora, and then fine-tune the models on downstream tasks (Devlin et al., 2018). The early pre-training techniques mainly focused on the natural language understanding tasks such as the GLUE benchmark (Wang et al., 2018) , and later it was gradually extended to the natural language generation tasks, e.g., NMT.

Recently, a prominent line of work has been proposed to improve NMT with pre-training. These techniques can be broadly classified into two categories. The first category usually uses pre-trained models as feature extractors of a source language, or initializes the encoder and decoder with pre-trained models separately (Lample and Conneau, 2019; Ren et al., 2019; Yang et al., 2020a; Zhu et al., 2020). For example, Lample and Conneau (2019) proposed a cross-lingual language model with a supervised translation language modeling objective, and used MLM or CLM to pre-train the encoder and decoder of NMT. However, the combined encoder-decoder model, where the cross-attention is randomly initialized, often does not work well because of the lack of semantic interfaces between the pre-trained encoder and decoder. There is also some work trying to leverage BERT-like pre-trained models for MT with an adapter (Guo et al., 2020) or an APT framework (Weng et al., 2020). The former defines additional layers in the pre-trained encoder and decoder during fine-tuning, while the last adopts a fusion mechanism or knowledge distillation to leverage knowledge in BERT for MT. Different from them, we enable the encoder and decoder to interact with a semantic interface during pre-training, and they can be connected directly for the MT fine-tuning without any other additional layers or training loss.

The second category methods pre-train a whole sequence-to-sequence model for NMT. MASS (Song et al., 2019) employed the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence. BART (Lewis et al., 2020) adopted a similar framework and trained the model as a denoising auto-encoder. mBART (Liu et al., 2020) trained BART model on large-scale monolingual corpora in many languages. Although the above work can pre-train the cross-attention of decoder, they are learned on monolingual denoising auto-encoding and cannot learn the corss-lingual transformation between source and target languages. There is also some work trying to explicitly introduce cross-lingual information in a code-switch way during the sequence-to-sequence pre-training, such as CSP (Yang et al., 2020b) and mRASP (Lin et al., 2020). However, their methods need a lexicon or phrase translation table, which is inferred from unsupervised cross-lingual embeddings. Therefore, they depend on the quality of the dictionary.

The most similar work to ours is probably the one of DALL·E and CLIP (Radford et al., 2020). DALL·E is a transformer language model that receives both the text and the image as a single stream of data. The core idea is to define the cross-modality interface of image and text, which can generate images from text descriptions. In this paper, to address the above limitations of pre-training methods for NMT, we attempt to define a cross-lingual semantic interface to connect the pre-trained encoder and decoder.

5 Conclusion

We propose SemFace, a better pre-training method for neural machine translation. The key point is to use a semantic interface to connect the pre-trained encoder and decoder. By defining this interface, we can pre-train the encoder and decoder separately with the same intermediate language-independent space. The cross-attention can also be pre-trained with our method so that we can naturally combine the pre-trained encoder and decoder for fine-tuning. We introduce and compare two semantic interfaces, e.g., CL-SemFace and VQ-SemFace, which leverage unsupervised cross-lingual embeddings and vector quantized embeddings as the intermediate interfaces respectively. Massive experiments on supervised and unsupervised NMT translation tasks show that our proposed SemFace obtains substantial improvements over the state-of-the-art baseline models. In the future, we will design and test more semantic interface types for extensions.

Acknowledgments

This work is supported in part by National Key R&D Program of China 2018AAA0102301, and NSFC 61925203.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075.

Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. *arXiv preprint arXiv:2010.06138*.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. *arXiv preprint arXiv:2010.03142*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2020. Learning transferable visual models from natural language supervision. *Image*, 2:T2.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Explicit cross-lingual pre-training for unsupervised machine translation. *arXiv preprint arXiv:1909.00180*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9266–9273.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020a. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9378–9385.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. Csp: Code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.