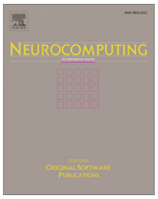
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Improving neural machine translation with latent features feedback

Yachao Li [a,b], Junhui Li [a], Min Zhang [a,*]

[a] School of Computer Science and Technology, Soochow University, Suzhou 215006, China
[b] Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou 730030, China

ARTICLE INFO

Article history:
Received 18 November 2020
Revised 6 June 2021
Accepted 9 August 2021
Available online 11 August 2021
Communicated by Zidong Wang

Keywords:
Neural machine translation
Latent features
Feedback
Transformer

ABSTRACT

Most state-of-the-art neural machine translation (NMT) models progressively encode feature representation in a bottom-up feed-forward fashion. This traditional encoding mechanism has no guidance from external signals. In computer vision tasks, the feedback connection plays a crucial role, particularly for understanding tasks. In this paper, we propose a simple but effective approach to learn latent feature representations explicitly from input sentences via a latent feature encoder (LFE), which are fed back to an NMT encoder via a top-down feedback mechanism. Through the feedback mechanism, the representations in one layer are influenced by representations of both lower and higher layers, resulting in a more effective encoding mechanism. Besides, to enhance the capability of the LFE in better capturing latent features from the source sentences, we pre-train the LFE via a Denoising Auto-Encoder (DAE) strategy. Experimentation on the large-scale WMT 2014 English-to-German and WMT 2017 Chinese-to-English translation tasks demonstrates that our proposed LFE, either pre-trained with the DAE or not, significantly outperforms the strong baseline.

1. Introduction

Neural machine translation [3,8,44,45] has achieved great success when learning solely from sentence-aligned parallel data with little or no external knowledge. Most state-of-the-art NMT models [45] consist of multiple encoder layers that compute progressively more invariant representations of an input sentence in an unsupervised way. Recent studies have demonstrated their strong capability of automatically learning certain feature representations from fine-grained to coarse-grained to well capture source sentence meaning at different levels of linguistic granularity. For instance, low layers may encode morphological representation [4] while middle layers may encode syntactic level representation [42], and high layers may encode semantic level representation [9,35]. These useful implicit features are helpful to further improve translation performance [49].

Biological evidence suggests that vision with scrutiny tasks, such as fine-grained categorizations, seem to require feedback along a reverse hierarchy [17]. Recent studies in cognition and computer vision [17,27,39] have shown that feedback plays a crucial role, particularly in understanding tasks. Global context information provides expectations, guiding our understanding of local inputs with ambiguity [31]. For instance, in the sentence "*I went to the* **bank** *with my good friend intending to withdraw some money.*", global semantic representations, i.e., top-down information of *intending to withdraw some money*, from the sentence will help disambiguate **bank** as a building rather than river bank. That is to say, top-down processing can leverage simultaneous global contextual information to resolve lexical ambiguity. In this paper, we explore the integration of the "bidirectional" features encoding into the traditional unidirectional bottom-up encoder via a top-down feedback context. In this way, the features of an NMT encoder layer are properly influenced by both fine-grained (low-level) and coarse-grained (high-level) feature representations.

To allow latent feature feedback, we first propose a latent feature encoder to effectively learn feature representations from source sentences by taking advantage of a vanilla *Transformer* encoder [45]. Then, we carefully feedback the learned latent features to an NMT encoder to properly guide the encoding process. As shown in Fig. 1(a), the LFE consists of multiple identical encoder layers and encodes different feature representations of source sentences in a bottom-up feed-forward fashion. Then the learned feature representations in the LFE are used to guide the NMT encoding in a top-down fashion. That is to say, to guide the *i*-th layer in the NMT encoder, we use feature representations of layers in LFE that are higher than *i*. Through this enhanced encoder with explicit guidance, we can capture more details of the source sentence. To make these learned feature representations more effective, we explore various ways to enhance the LFE as follows:

[*] Corresponding author.
E-mail address: minzhang@suda.edu.cn (M. Zhang).

https://doi.org/10.1016/j.neucom.2021.08.019
0925-2312/© 2021 Elsevier B.V. All rights reserved.

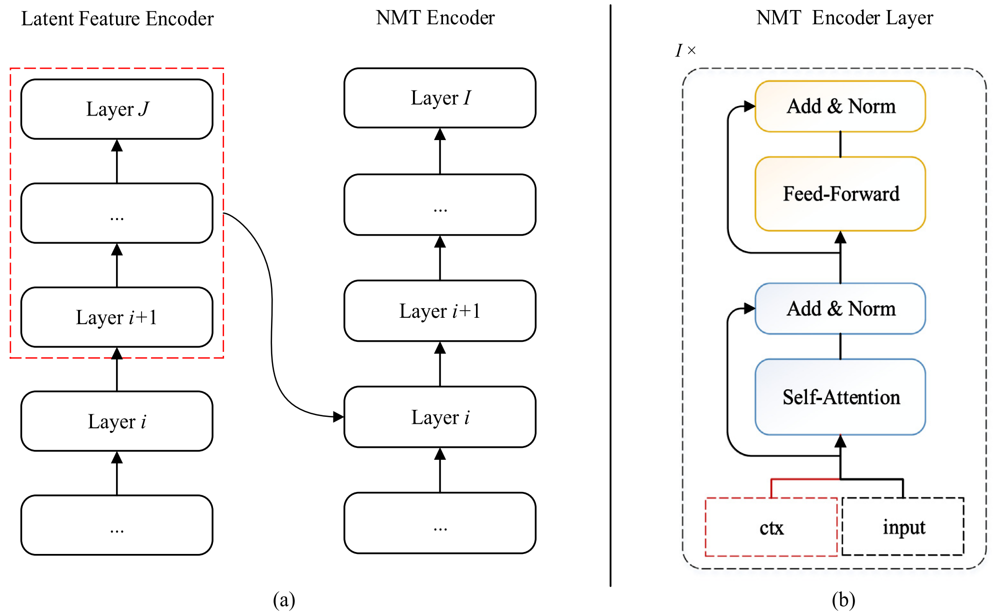


Fig. 1. (a) Our proposed NMT encoder model which consists of a latent feature encoder through top-down feedback connections, (b) an updated NMT encoder layer with feedback representation from LFE.

- Joint-trained LFE. We train the LFE together with other parts of an NMT model by sharing a loss function.
- Pre-trained LFE. We first pre-train the LFE via a Denoising Auto-Encoder strategy and then fix the LFE during NMT training.
- Pre-trained and fine-tuned LFE. Different from above, we fine-tune the LFE together with other parts of the NMT model during NMT training.

Experimental studies on two large-scale WMT English-to-German and Chinese-to-English translation tasks show that our latent feature feedback mechanism significantly improves the translation performance. For example, the proposed pre-trained LFE with fine-tuning achieves significant gains of 1.64 and 1.05 BLEU scores over Transformer on the two translation tasks, respectively. We also provide extensive analyses to explain how our latent feature feedback mechanism improves the translation quality.

2. NMT encoder with latent feature feedback

We take the state-of-the-art seq2seq model, i.e., the *Transformer* [45] as our baseline. In the following, we will describe the details of a *Transformer* encoder layer, which will be used for the NMT encoder and the proposed latent feature encoder.

A conventional Transformer encoder layer (also called a Transformer encoding block) consists of two sub-layers: a multi-head self-attention sub-layer followed by a position-wise feed-forward network sub-layer. Such an encoder layer takes a sequence $x = (x_1, \ldots, x_n)$ of n elements as input and computes a new sequence $h = (h_1, \ldots, h_n)$ of the same length. First, intermediate state $z = (z_1, \ldots, z_n)$ is obtained through the multi-head self-attention sub-layer:

$$z = LN\Big(SA(W^Q x, W^K x, W^V x) + x\Big), \tag{1}$$

where $x, z \in \mathbb{R}^{n \times d_h}$, and d_h is the hidden state size. $LN(\cdot)$ denotes layer normalization [2], and $SA(\cdot)$ represents the multi-head self-attention network that linearly maps x into query, key-value sets via matrices $W^Q, W^K, W^V \in \mathbb{R}^{d_h \times d_h}$. Then, the intermediate state z is fed into the feed-forward network sub-layer which outputs h by:

$$h = LN(FFN(z) + z), \tag{2}$$

where $h \in \mathbb{R}^{n \times d_h}, FFN(\cdot)$ represents the feed-forward network. To alleviate gradient problems [5], residual connections [15] and layer normalization [2] are applied in both sub-layers.

The above conventional NMT encoder adopts a bottom-up feed-forward information processing pathway, which progressively encodes feature representations from fine-granularity in lower layers to coarse-granularity in higher layers. However, this largely ignores the practical experience that feedback plays a crucial role in understanding tasks [17,27,39]. On this basis, our purpose is to inform the NMT encoder, especially its low encoder layers, of the coarser feature representations hidden in source sentences.

Fig. 1(a) illustrates the proposed NMT encoder model which consists of an LFE and (b) an NMT encoder with updated encoder layers. Here the latent feature representations learned by the LFE are fed back to the updated Transformer encoding layers in the NMT encoder to guide the encoding more effectively.

2.1. Latent feature encoder

Previous studies demonstrate that a seq2seq NMT encoder, especially Transformer-based, has the strong capability of automatically learning feature representations that capture source sentence meaning [42,4,34]. Therefore, it is reasonable to use an existing seq2seq NMT encoder as our LFE. In particular, we choose an original Transformer encoder with J Transformer encoder layers as the LFE, as shown in the left part in Fig. 1(a).

We use s^i to indicate the output of the i-th $(1 \leqslant i \leqslant J)$ encoder layer in the LFE. The top encoder layer in the LFE, not surprisingly, captures sentence-level semantic features, while the lower layer states encode syntactic and morphological feature representations. In the next section, we will introduce how to feed these captured useful features representation to the NMT encoder layers.

2.2. NMT encoder with top-down feedback

Inspired by feedback connections in computer vision [17,39,27], which allow the information in one layer to be implicitly influenced by those above and below, and it can also significantly improve the ability of the encoder to capture the details of the source inputs. We propose a *rectified self-attention* sub-layer to enhance the original NMT encoder. Besides the input sequence x, the Transformer encoder layer in the NMT encoder is equipped with latent feature representations ctx learned by LFE, as shown in Fig. 1(b). Note that x and ctx are from previous layer outputs, so both of them are normalized by $LN(\cdot)$ operation. We first merge them into x' by:

$$x' = x + ctx, \tag{3}$$

and then update Eq. (1) accordingly to propagate latent feature representations to the self-attention sub-layer by:

$$z = LN\Big(SA(W^Q x', W^K x', W^V x') + x\Big). \tag{4}$$

In this way, the latent feature representations can be selectively received in the *rectified self-attention* sub-layer and the subsequent operations. Moreover, the feature representations of LFE can be used as rectified features to guide the representation of the current layer. Thus, the proposed NMT encoder allows feature representations of the current layer to be naturally influenced by those above (representations from the higher layers of LFE) and below (representations from the lower layers of NMT encoder), see Section 5.7 for more discussion. As shown in Fig. 1(b), we keep the feed-forward network sub-layer unchanged. Note that compared to the original Transformer encoder layer, the updated Transformer encoder layer does not introduce any additional parameters.

Since the NMT encoder consists of I updated Transformer layers, we use superscript i to indicate the i-th updated Transformer layer. Given an LFE with J layers and an NMT encoder with I layers $(I \leqslant J)$, we now explore the ways of employing latent feature representations ctx, as shown in Eq. (3). For the i-th $(1 \leqslant i \leqslant I)$ layer in the NMT encoder, we explore ctx^i from LFE in different ways, investigating the effect of latent feature representations from different layers.

As shown in Fig. 1, we add top-down feedback connections to the NMT encoder, and combine feature representations of internal states in LFE, to capture different types of lexical, clause-level, and sentence-level information. For the i-th NMT encoder layer, we define a function that fuses feature representations from all the higher layers of the LFE, i.e.:

$$ctx^i = f\left(s^{i+1}, \ldots, s^J\right), \tag{5}$$

where $f(\cdot)$ is a fuse function and we use mean pooling [19] in all our experiments.

As in Eq. (5), we extract feature representations from higher layers of LFE (also higher than the current NMT encoder layer) to the current NMT encoder layer, thus we refer to it as *top-down feedback*. This is based on the consideration that both the LFE and NMT encoder takes source sentence embedding as input. On this basis, the higher an encoder layer is, the richer information it captures from the input. Therefore, to better guide the encoding process of the current layer, we use richer information by fusing all the feature representations from higher layers of LFE.[1] Specifically, we set ctx^I for the top I-th NMT encoder layer as zero. Note that for the i-th layer of NMT encoder, there are other various ways to define useful ctx^i from the LFE. For example, we can simply view it as the final output of the LFE, i.e., $ctx^i = s^J$. However, our preliminary experimental results show that our fusing method outperforms it with negligible computing cost.

Although some related work has explored different latent features to improve translation performance. For instance, the work of [6,29,52] proposed top-down encoding to enhance the NMT performance by incorporating explicit syntactic or context information, however, our proposed latent features are automatically obtained from monolingual data. Moreover, recent studies show that the fusion of encoding layers is beneficial for NMT [11,12,48], however, these methods only fuse the low layer features into the current layer. Partly inspired by this work, the purpose of our approach is to explicitly feedback the higher latent features to the lower layers, which can be used to regulate the source input encoding and resulting in better source language representations.

3. Pre-training of latent feature encoder via Denoising AutoEncoder

The above end-to-end architecture uses cross-entropy losses on the target sentences to train the whole model, including the LFE. That is to say, the LFE is trained to reduce the loss of the target translations. To enhance the capability of the LFE in capturing latent features from the source sentences, we pre-train the LFE explicitly to recover the source sentences themselves via autoencoding.

Autoencoding of sentences can be naturally viewed as a seq2seq task. To avoid degenerating autoencoding as a trivial copying task and enable the encoder, i.e., the IFE, effectively learn the useful linguistic features of its input sentences, we follow recent advances in unsupervised neural machines translation [1,26], to adopt the Denoising AutoEncoders (DAE) strategy [46], which attempts to reconstruct the original version of a corrupted input sentence.

Given an input sentence $s = (s_1, \cdots, s_n)$ with n words, we add three different types of noise to s, and get its corrupted version s'. First, we slightly shuffle the sentence by performing N_{swap} swaps, in each of which we randomly select two words. Second, we randomly select $n \times N_{drop}$ words and drop them from the sentence. Finally, we randomly mask $n \times N_{mask}$ words with a placeholder token $<pad>$ to predict the original vocabulary of the masked word based only on its context. In our experiment, we set N_{swap} as 3, N_{drop} as 0.1, and N_{mask} as 0.2.[2] Fig. 2 shows an example of an input sentence s and its corrupted version s'.

While the LFE transfers the input s' into its hidden states, we use a Transformer-based decoder to reconstruct s. Therefore, the DAE adopted here is a standard Transformer structure. Note that we do not initialize the NMT model with the parameters on the decoder side of the pre-trained model.

When integrating the pre-trained LFE into the NMT model, as shown in Fig. 1(a), we can either fix or fine-tune the LFE during NMT training. Here, we do not add any noise to the input sentences of LFE. Moreover, since the LFE is independently pre-trained, we do not share its word embedding with the NMT encoder, though the two encoders share one vocabulary.

Pre-training is an effective approach to obtain useful latent features from monolingual or bi-bilingual data, which can be used in downstream tasks [10]. BERT [10] and XLM [25] adopt a masked language model for pre-training. For sequence-to-sequence pre-training models, MASS [43] masks the source input words, while

[1] Although we can also choose the latent features from the lower layers in LFE to guide the NMT encoder process. However, in our preliminary experiments, these latent features from lower layers have a limited effect on translation performance. This also shows that our method is essentially different from the traditional feature fusion methods[49,51].

[2] The values are the default settings in the source codehttps://github.com/facebookresearch/XLM and we do not tune them.

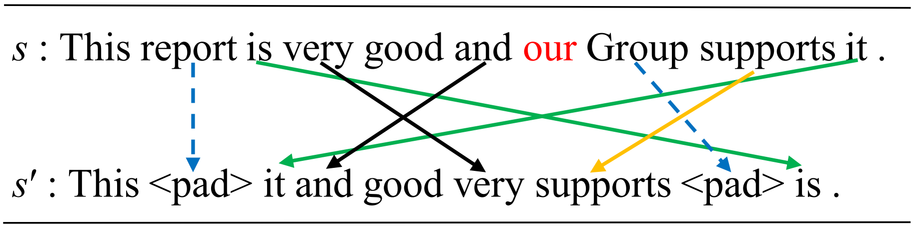

s : This report is very good and our Group supports it .
s′ : This <pad> it and good very supports <pad> is .

Fig. 2. An example of an input sentence s and its corrupted version s'. The red word indicates the dropped word, and the dotted lines indicate the masked operations, while the solid lines are for swapped operations.

the decoder predicts the masked words. Our approach shares a similar idea as BART [28], while the major difference is that the parameters of the decoder side are not used in our proposed approach. For the reason that we not only pay attention to the pre-trained source representations but also to reduce the influence of other parameters.

4. Experimentation

To systematically evaluate our approach, we carry out various experiments on two large-scale translation tasks: WMT 2014 English-to-German (En-De) and WMT 2017 Chinese-to-English (Zh-En). These are two mainstream machine translation data sets.

4.1. Experimental settings

Dataset For En-De translation, the data is from WMT 2014,[3] which consists of 4.58 M sentence pairs in the training set. We use news-test-2013 as the development set and news-test-2014 as the test set. For Zh-En translation, the data is from WMT 2017,[4] which consists of 20.50 M sentence pairs. We use news-dev-2017 as the development set and news-test-2017 as the test set. We use the 4-gram NIST BLEU score [32] as the evaluation metric.

Both the En-De and the Zh-En datasets are tokenized by the Moses scripts. We segment words into sub-word units by byte-pair encoding (BPE) [40] with 32 K operations on both the source and the target languages. After that, we filter out long sentence pairs by limiting the maximum sentence length to 60 words (tokens) for En-De and 50 words (tokens) for Zh-En translation, respectively. For all experiments, we share bilingual vocabulary to reduce the computation cost.

Model Details We use *OpenNMT* [22] as the implementation of the Transformer seq2seq model.[5] We follow [45] to set the configurations and train the models. For all the experiments, the sizes of source embedding, target embedding, and hidden states are all set to 512. The number of layers in the LFE, NMT encoder, and decoder is 6 while the filter size in FFN is 2048, and the multi-head attention has 8 individual heads. For training, we use Adam [21] to train various NMT models with an initial learning rate of 2.0, the warmup step of 8000, and the mini-batch token size of 4096. We use the learning rate decay policy proposed by [45] with dropout [16] of 0.1. For inferring, the beam size is set to 6 for En-De translations and 4 for Zh-En translations. All the translation models are trained on four GPUs with gradients accumulation of 2.

To train the autoencoders for the two source languages, English and Chinese, we use Transformer with the same settings as their corresponding En-De and Zh-En translations, except that we employ the vocabulary from their corresponding En-De and Zh-En translations, and that we set the mini-batch token size as 4096 with gradient accumulation of 1 to speed up the training process.[6]

4.2. Experimental results

En-De Translation Table 1 shows the performance of En-De translation measured in BLEU. It shows that all the three systems with LFE improve the translation performance over the baseline Transformer on the test set, suggesting the NMT systems benefit from the latent feature feedback learned by the LFE. From Table 1, we observe:

- The *+ LFE (joint-trained)* (model #2) achieves a significant improvement of 1.13 BLEU scores over the baseline, suggesting that the joint training strategy is very effective for the LFE to learn latent feature representations from source sentences and that the representations are helpful for NMT encoding.
- The *+ LFE (pre-trained)* (model #3) achieves higher performance than the one with joint trained LFE. This implies that the pre-trained LFE is better than the joint-learned LFE in capturing latent feature representations from source sentences. This is consistent with findings in Section 5.7 that we tend to recover more linguistic information from the pre-trained LFE than from the joint-learned LFE.
- The *+ LFE (pre-trained + fine-tuning)* (model #4) achieves the highest performance with a 29.27 BLEU score on the test set, with 0.24 improvement over the one without fine-tuning, suggesting that fine-tuning the LFE is helpful for translation.

It can be seen from the above that the LFE proposed by us can enhance the performance of the NMT encoder. Moreover, the pre-trained LFE has better performance.

Zh-En Translation Table 2 presents the performance on the Zh-En translation task. The results show that the performance trend over the proposed systems is in line with that of En-De translation. For instance, the best system (i.e., model #4) significantly outperforms the baseline system with 1.05 BLEU improvement on the test set. Such improvement is very promising, considering that the models are trained on a large-scale dataset with 20.50 M sentence pairs, from which the encoder of the baseline may have already well captured rich linguistic information for translation.

4.3. Training speed

It is not surprising that introducing LFE slightly slows down the training speed. When running on four GPUs GeForce GTX 1080, the baseline runs at 0.97 s per iteration while the improved systems only increase the training time by about 12% to cater to the LFE.

5. Analysis and discussion

[3] http://www.statmt.org/wmt14/translation-task.html

[4] http://www.statmt.org/wmt17/translation-task.html

[5] https://github.com/OpenNMT/OpenNMT-py

[6] We speculate that even better translation results would be obtained by tuning the architecture of this autoencoding task.

Table 1
BLEU scores of En-De translation on the news-test-2014 test set. Hereafter, †/‡: significant over the baseline *Transformer-Base* at 0.05/ 0.01, tested by bootstrap resampling [23]. "#Para." indicates the number of parameters (million) while "Speed" denotes the training speed (second/iteration).

#	Model	#Para.	Speed	BLEU
1	Transformer-Base	76.0 M	0.97	27.63
2	+ LFE (joint-trained)	98.1 M	1.09	28.76 ‡
3	+ LFE (pre-trained)	98.1 M	1.09	29.03 ‡
4	+ LFE (pre-trained + fine-tuned)	114.5 M	1.09	**29.27** ‡

Table 2
BLEU scores of Zh-En translation on the news-test-2017 test set.

#	Model	#Para.	Speed	BLEU
1	Transformer-Base	79.0 M	0.98	23.34
2	+ LFE (joint-trained)	101.1 M	1.10	23.95 ‡
3	+ LFE (pre-trained)	101.1 M	1.09	24.12 ‡
4	+ LFE (pre-trained + fine-tuned)	114.5 M	1.10	**24.39** ‡

In this section, we carry out additional experiments on En-De translation and further explore more on how our proposed feature feedback from the LFE helps translation.

5.1. Performance of AutoEncoding

To evaluate the performance of autoencoding, we use BLEU as a metric to assess how well the autoencoder reconstructs the original version of a corrupted source sentence. For comparison, we use the original sentences as input to assess how much the autoencoder changes them. In both settings, we compute BLEU scores against the original sentences.

Table 3 shows the BLEU scores of recovering original sentences. The high performance of recovering from corrupted version (e.g., 50.00 in BLEU scores) suggests that the autoencoder can effectively capture useful latent features from the input sentences, while the even higher performance of recovering from the original version (e.g., 79.89) suggests that the autoencoder would not change the meaning of input sentences.

5.2. Comparison with deep NMT encoder

It is well known that increasing encoder layers will indeed improve translation performance [47]. To verify what the major contribution of our proposed method is, we compare our approach with the deep NMT model with the same number of encoder layers.

As shown in Table 4, the system #2 and #3 have the same encoder layers (12 layers in total), however, our proposed LFE with joint-training (#3) outperforms the deep NMT encoder with 12 layers. This indicates that the useful latent features fed back from LFE can improve the translation quality. Compared with the 12-layer encoder deep NMT model (#3), our proposed method (#4) achieves a significant improvement over the strong baseline. This shows that the LFE with pre-training and fine-tuning can better improve translation performance.

Table 3
BLEU scores of the autoencoder on the En-De translation test set.

#	Input	BLEU
1	Corrupted	50.00
2	Original	79.89

Table 4
Comparison between our proposed encoder and deep NMT encoder on translation.

#	Model	Param.	BLEU
1	Transformer-Base	76.0 M	27.63
2	Transformer-12	98.1 M	28.37
3	+LFE (joint-trained)	98.1 M	28.76
4	+LFE (pre-trained + fine-tuned)	114.5 M	**29.27** ‡

5.3. Comparison between LFE and NMT encoder

To compare the translation performance of LFE and NMT encoder, we choose joint-trained LFE as representative (model #2 in Table 1) and take LFE (#2) and NMT (#3) outputs as source languages representation respectively, to compare their translation performance.

As shown in Table 5, the experimental performance of the LFE encoder (#2) is similar to that of the baseline system (#1), because LFE is a vanilla Transformer encoder. The NMT encoder (#3) significantly improves the translation performance, although models #2 and #3 have the same parameter size. It can be seen that the NMT encoder can be properly guided by LFE features via a feedback mechanism, which significantly improves translation performance.

5.4. Effect on lower and higher NMT encoder layers

As a representative, we choose our system with pre-trained LFE which is not further fine-tuned in NMT training, i.e., model #3 in Table 1. As shown in Fig. 1, our model uniformly equips all NMT encoder layers with the latent feature representations. To investigate which layers of the NMT encoder benefit more from latent feature representations, we add latent feature representations *ctx* to the 1–3 and the 4–6 layers of the NMT encoder respectively and compare their contribution to translation improvement.

As shown in Table 6, both the lower encoder layers (e.g., 1–3) and the higher layers (e.g., 4–6) benefit from the latent feature representations. As our expectation, the performance trend suggests

Table 5
Evaluation of output of LFE and NMT encoder on translation.

#	Model	Param.	BLEU
1	Transformer-Base	76.0	27.63
2	+LEF encoder	98.1	27.69
3	+NMT encoder	98.1	**28.76** ‡

Table 6
Evaluation of applying latent feature feedback with different encoder layers. "#Layer" indicates which encoder layers to apply.

#	Model	#Layer	BLEU
1	Transformer-Base	n/a	27.63
2	+LFE (pre-trained)	1–3	28.82 ‡
3		4–6	28.15 †
4		1–6	**29.03** ‡

that the lower NMT encoder layers benefit more than the higher layers, indicating that the lower layers are more sensible to be guided from the sentence-level information. This is consistent with the observations of [51], in which they found that the lower encoder layers require more sentence-level context than higher layers.

5.5. Comparison with related pre-training models

We choose our best system, i.e., model #4 in Table 1 to compare the performance with related pre-trained models. As shown in Table 7, we train LFE by BERT (#2), MASS (#3), or our proposed approach (#4), and then compare their performance on translation.

As shown in Table 7, the model #2 improves the translation performance, indicating that the pre-trained LFE by BERT task helps translation. We also find that the performance of model #3 is better than that of model #2, which indicates that the sequence-to-sequence pre-training task can better capture useful features suitable for NMT. Our proposed approach (model #4) achieves the best experimental results, which shows that our proposed LFE pre-training approach can better capture useful latent features.

5.6. Shared parameters between LFE and NMT encoder

In NMT, increasing the size of parameters will indeed promote translation performance [45]. In addition, adding extra parameters will bring noise information. To make a more fair comparison with the baseline system, we share the parameters between the LFE and NMT encoder. Therefore, the parameter sizes of all systems are the same. In this case, the main difference between the proposed models is whether the LFE has undergone pre-trained.

As shown in Table 8, compared with the results of Table 1, sharing parameters have a slight effect on the translation performance. On the whole, our proposed approaches (#3) significantly improve the translation quality, even though both of them have the same parameter size as the baseline system. This shows that our proposed LFE indeed improves the translation quality.

Table 7
Comparison between our proposed pre-training approach and related work. "+BERT" and "+MASS" indicate the LFE trained by BERT or MASS task.

#	Model	Param.	BLEU
1	Transformer-Base	76.0 M	27.63
2	+BERT (pre-trained + fine-tuned)	114.5 M	28.21 †
3	+MASS (pre-trained + fine-tuned)	114.5 M	28.96 ‡
4	+LFE (pre-trained + fine-tuned)	114.5 M	**29.27** ‡

Table 8
Experimental results of our approach with sharing parameters between the NMT encoder and LFE.

#	**Model**	Param.	**BLEU**
1	Transformer-Base	76.0 M	27.63
2	+LFE (joint-trained)	76.0 M	28.53 ‡
3	+LFE (pre-trained + fine-tuned)	76.0 M	**29.06** ‡

5.7. Linguistic analysis

In this section, we implement probing tasks to gain linguistic insights into what kind of linguistic features are present in the latent feature representations. We follow [9] to evaluate linguistic knowledge embedded in following types of representations: 1) the layers of NMT encoder in the baseline system (as shown in Fig. 3); 2) the output of the proposed LFEs (as shown in Fig. 4); 3) the top layer of NMT encoder with feature feedback (as shown in Fig. 5)); and 4) the first layer of NMT encoder enhanced with feature feedback (as shown in Fig. 6).

The probing tasks can be viewed as classification problems that focus on various linguistic properties of source sentences. Specifically, the 10 probing tasks aim to recover source-side linguistic information, which further falls into three categories: surface information (the sentence length (SeLe) and word content (WC) tasks), syntactic information (the tree depth (TrDep), top constituent (ToCo) and bigram shift (BShif) tasks) and semantic information (the tense (Tense), subject number (SubN), object number (ObjN), semantic odd man out (SoMo), and coordination inversion (CoIn) tasks). For all tasks, we use the default settings, as well as the provided dataset.[7]

(1) Analysis of Baseline Model. As shown in Fig. 3, the representation of lower layers contains more surface information while higher layers encode more syntactic and semantic information. This trend is consistent with previous findings in [38,49,20,13]. These findings also indicate that the encoding process of the NMT encoder is to continuously capture the high-level information (syntax and semantic features) and correspondingly remove the low-level (surface features) features.

(2) Analysis of the top layer of LFE Models. For the LFEs (as shown in Fig. 4), we see that the pre-trained LFE, either fine-tuned or not, captures more information than the joint-trained LFE over all probing tasks except WC and TrDep. Moreover, fine-tuning the LFE seems to have a slight influence on the performance of all probing tasks except WC. Finally, compared to the top layer of the baseline encoder, the pre-trained LFEs tend to contain more linguistic information (except task WC and TrDep). This useful linguistic information will guide the encoder process of NMT.

(3) Analysis of the top layer of NMT encoder enhanced with LFE. For the top NMT encoder layers after integrating the proposed LFEs (as shown in Fig. 5), there is no obvious sign indicating which one outperforms the other two. Compared to the top layer of the baseline encoder (Transformer-L6), however, we observe that with the latent features feedback, the semantic information is enhanced. While our NMT systems with LFE trend to capture more context word (WC) information (+LFE (pre-trained and + LFE(pre-trained + fine-tuned) in WC task). Since content words play an important role for NMT [7]. This shows that both the semantic features and lexical features are important for NMT. However, the original Transformer model tends to encode more syntactic and semantic features while losing the low-level features, such as lexical features. This indicates that our approach can selectively encode useful linguistic features of source input, and the overall trend of our encoder is different from that of baseline encoders.

(4) Analysis of the first layer of NMT encoder enhanced with LFE. We take probing tasks on the first layer of the NMT encoder

[7] https://github.com/facebookresearch/SentEval

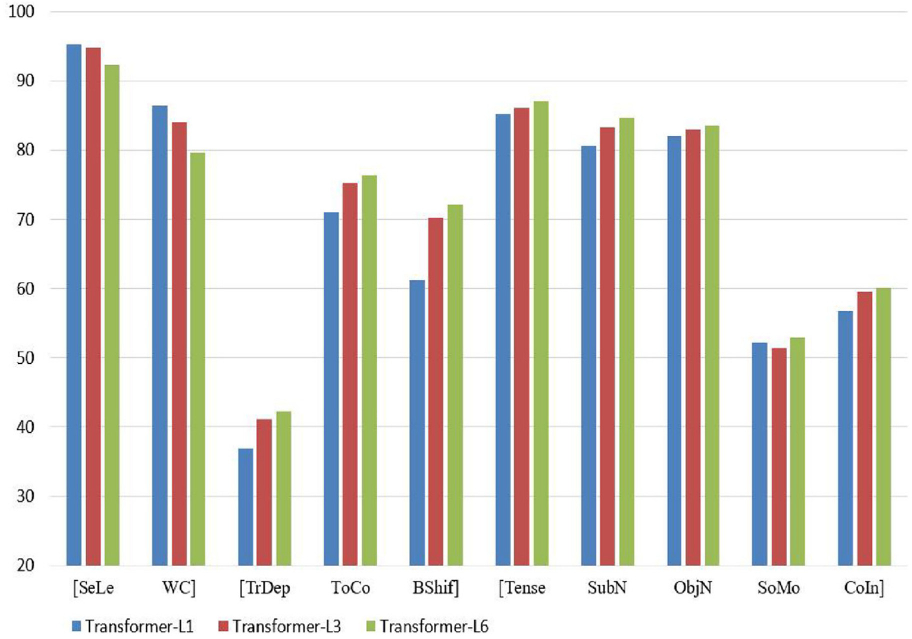


Fig. 3. Performance of the probing tasks based on encoder layers of the baseline system with 6 encoder layers. "Transformer-*LX*" denotes the *X*-th encoder layer output. The y-axis represents the accuracy (%).

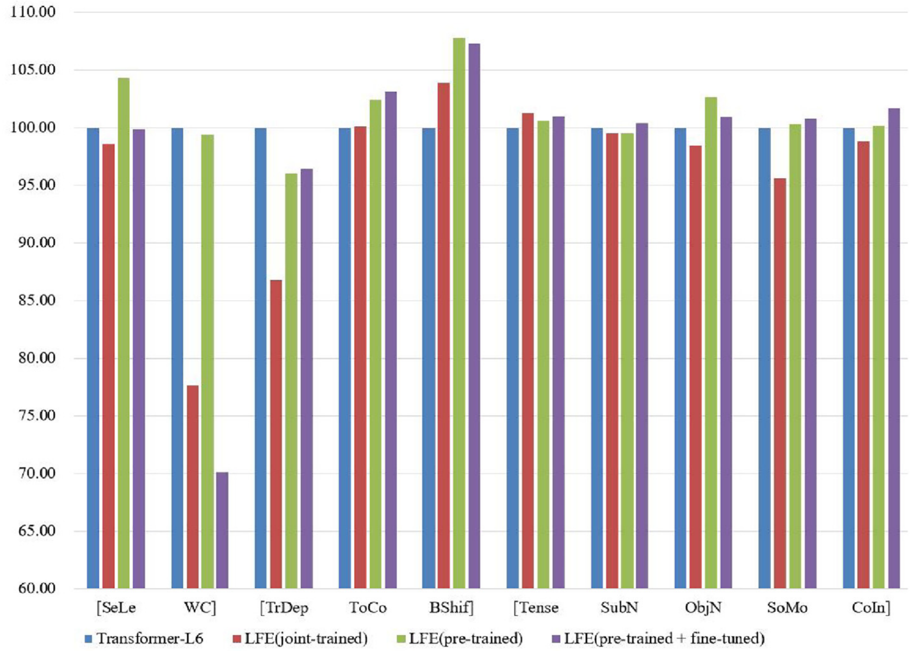


Fig. 4. Performance of the probing tasks based on the top layer of the baseline system (Transformer-L6) and the proposed LFE models. For each probing task, we scale the performance of the top layer of the baseline system as 1.0 for better illustration. The y-axis represents the percentage.

to explore the influence of latent features. As shown in Fig. 6, after integrating latent features, the surface information of the first layer is reduced quickly, which indicates that the high-level features are helpful for features encoding. Compared with the first layer of the baseline system (Transformer-L1), the syntactic and semantic features are significantly improved after integrating the latent features from the LFE. Among them, the syntactic features increase more obviously. This shows that our proposed encoder can capture more syntactic and semantic features at the low layers. It also shows that the high-level features of LFE can significantly modulate the features encoding in low layers of the NMT encoder.

(5) To sum up, we can conclude that the original Transformer model is to continuously capture high-level features such as syntactic and semantic information while reducing the surface information. Compared with the baseline model, our proposed approach can selectively capture useful features with the guidance of LFE, such as WC. These findings are consistent with the previous work, in which they found that WC is an important feature for NMT [53].

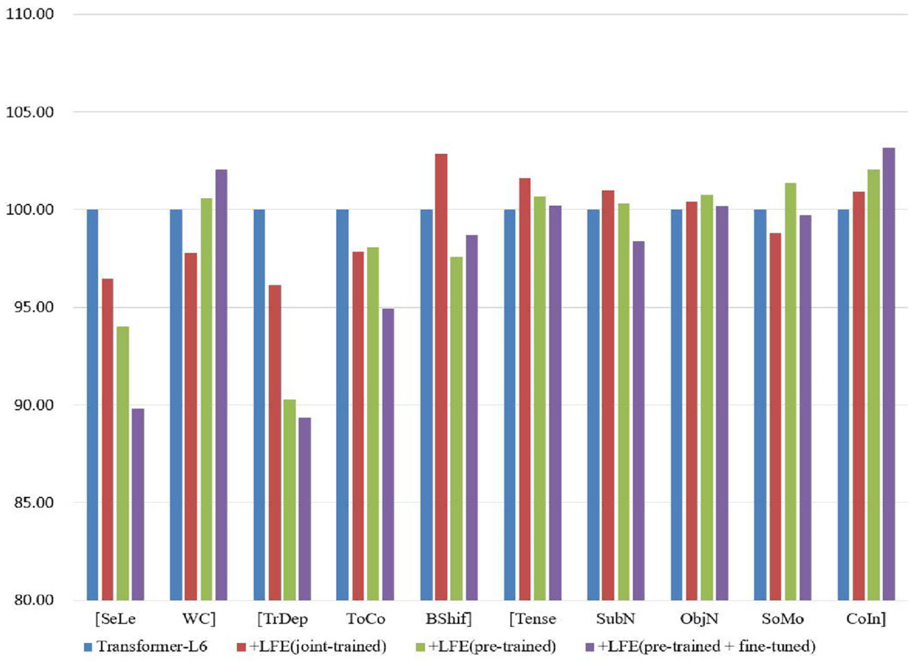


Fig. 5. Performance of the probing tasks based on the top layer of the baseline system (Transformer-L6) and the proposed NMT encoder enhanced with LFE. For each probing task, we scale the performance of the top layer of the baseline system as 1.0 for better illustration. The y-axis represents the percentage.

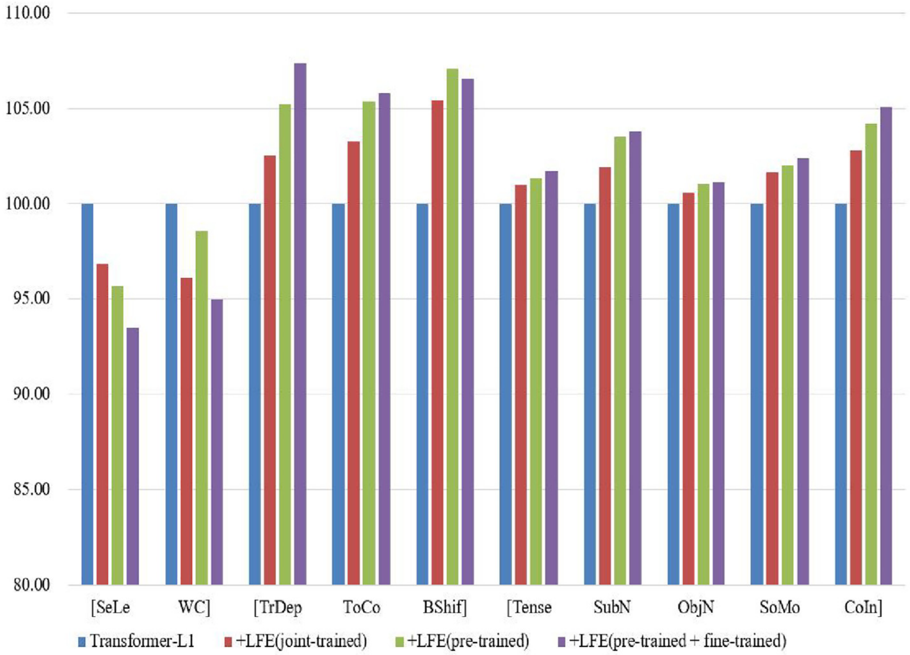


Fig. 6. Performance of the probing tasks based on the first layer of the baseline system (Transformer-L1) and the proposed NMT encoder enhanced with LFE. For each probing task, we scale the performance of the first layer of the baseline system as 1.0 for better illustration. The y-axis represents the percentage.

5.8. Effect on long sentences

As translating long sentences usually requires the NMT encoders to be capable of capturing the long-distance dependencies, we conjecture that long sentences benefit more from sentence-level latent feature representations. Following [41,55], we partition source sentences in the test set to different groups by their lengths and compute their BLEU scores respectively.

Fig. 7 presents the BLEU scores. The performance indicates that incorporating latent feature representations outperforms the baseline overall sentence lengths. We also observe that the performance gap between the baseline and our approach increases when input sentences become longer, revealing that long sentences benefit more from latent feature representations and that the baseline is far from capturing deep linguistic details of long sentences.

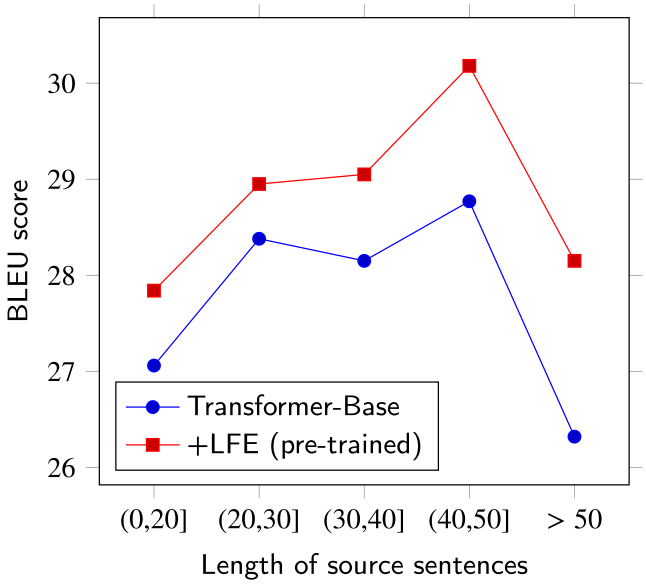


Fig. 7. BLEU scores on the test sets concerning the lengths of the input sentences.

6. Related work

We discuss related work from the following two perspectives.

Feedback Connections. In computer vision, feedback plays a crucial role in many classic models [17,27]. For example, vision scrutiny tasks, including fine-grained categorization [24] or detailed spatial manipulations [18], appear to need feedback connections. Hierarchical probabilistic models allow random variables in one layer to be naturally influenced by those above and below [17]. Part models [14] allow a face object to influence by the eye part through top-down feedback. Top-down neural saliency methods [39] can find important regions given a high-level semantic task. Inspired by these studies, we propose a top-down feedback mechanism in the NMT encoder, which can make the encoding process of the current layer be influenced by both the lower layers and the higher layers simultaneously. Experimental results show that the automatic learning of useful latent linguistic features via feedback connections significantly improves the translation quality.

Unsupervised Pre-training Approaches.. Unsupervised pre-training approaches have been widely proposed in various areas. [30,33] propose to learn word embeddings, which could be used to initialize word embeddings in downstream tasks. Recently, pre-trained representations [34,36,10] adopt self-supervised learning and yield powerful language models that considerably outperform the prior art. [34] propose ELMo to learn contextualized representations using bidirectional language model. OpenAI [36,37] propose GPT-2 which is a left-to-right Transformer language model. BERT [10] applies bidirectional training of Transformer, a popular attention model, to language modeling. Unlike the above encoder-only (e.g., ELMo, BERT) or decoder-only (e.g., GPT) pre-training approaches, sequence-to-sequence pre-training approaches have also been widely concerned, such as MASS [43] and BART [28]. Inspired by these pre-training approaches, in this paper we use DAE to pre-train the proposed LFE, to better capture the features of source sentences.

7. Conclusion

In this paper, we present a latent feature feedback mechanism for neural machine translation. Specifically, we propose a latent feature encoder to capture latent feature representations from input sentences, which is fed back to the NMT encoder via a top-down feedback mechanism. To make latent feature representations more effective, we explore the joint-trained and pre-trained LFE to better capture the latent features of source sentences. Experimental studies on English-to-German and Chinese-to-English translation tasks show that our proposed approach achieves significant improvement over the strong Transformer model. We have also analyzed the translation behavior of our improved systems against the state-of-the-art NMT baseline system from several perspectives.

From the linguistic analysis, we note that there is still room for the LFE to capture more useful latent features. To build better LFE, in future work we would like to explore the ways of distilling knowledge learned in pre-trained models, like BERT. Moreover, we would like to validate the latent feature feedback mechanism in other neural architectures, such as RNN-based and CNN-based models. Meanwhile, since Transformer-based encoders have widely been used in other NLP applications to encode input sentences [50,54,56], we would like to validate the latent feature feedback mechanism for them. Another promising direction is to extract more effective latent features from large-scale monolingual data for NMT or other tasks.

CRediT authorship contribution statement

Yachao Li: Conceptualization, Methodology, Writing - original draft. **Junhui Li:** Methodology, Writing - review & editing. **Min Zhang:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank anonymous reviewers for providing helpful comments. This work was supported by the National Natural Science Foundation of China (Grant No. 62036004, 61876120).

References

[1] M. Artetxe, G. Labaka, E. Agirre, K. Cho, Unsupervised neural machine translation, in: Proceedings of ICLR, 2018..
[2] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization. Computing Research Repository, 2016. arXiv:1607.06450. https://arxiv.org/abs/1607.06450..
[3] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of ICLR, 2015..
[4] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, J. Glass, What do neural machine translation models learn about morphology?, in: Proceedings of ACL 2017, Vancouver, Canada, 2017. pp. 861–872. https://www.aclweb.org/anthology/P17-1080, doi: 10.18653/v1/P17-1080..
[5] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Transactions on Neural Networks 5 (1994) 157–166.
[6] H. Chen, S. Huang, D. Chiang, J. Chen, Improved neural machine translation with a syntax-aware encoder and decoder, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1936–1945, https://doi.org/10.18653/v1/P17-1177.
[7] K. Chen, R. Wang, M. Utiyama, E. Sumita, Content word aware neural machine translation, in: Proceedings of ACL 2020, Association for Computational Linguistics, Online, 2020. pp. 358–364. https://www.aclweb.org/anthology/2020.acl-main.34, doi: 10.18653/v1/2020.acl-main.34..
[8] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, Proceedings of EMNLP (2014) 1724–1734.
[9] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic

properties, in: Proceedings of ACL 2018, Melbourne, Australia, 2018. pp. 2126–2136. https://www.aclweb.org/anthology/P18-1198, doi: 10.18653/v1/P18-1198..

[10] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in, 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), 2019, pp. 4171–4186, https://doi.org/10.18653/v1/N19-1423.

[11] Z.Y. Dou, Z. Tu, X. Wang, S. Shi, T. Zhang, Exploiting deep representations for neural machine translation, in: Proceedings of EMNLP 2018, Brussels, Belgium, 2018. pp. 4253–4262. https://www.aclweb.org/anthology/D18-1457, doi: 10.18653/v1/D18-1457..

[12] Z.Y. Dou, Z. Tu, X. Wang, L. Wang, S. Shi, T. Zhang, Dynamic layer aggregation for neural machine translation with routing-by-agreement, in: Proceedings of AAAI, 2019..

[13] K. Ethayarajh, How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings, in: Proceedings of the EMNLP-IJCNLP 2019, Association for Computational Linguistics, Hong Kong, China, 2019. pp. 55–65. https://www.aclweb.org/anthology/D19-1006, doi: 10.18653/v1/D19-1006..

[14] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1627–1645.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of CVPR (2016) 770–778.

[16] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors. Computing Research Repository, 2012 arXiv:1207.0580. https://arxiv.org/abs/1207.0580..

[17] P. Hu, D. Ramanan, Bottom-up and top-down reasoning with hierarchical rectified gaussians, Proceedings of CVPR 2016 (2016) 5600–5609, http://openaccess.thecvf.com/content_cvpr_2016/papers/Hu_Bottom-Up_and_Top-Down_CVPR_2016_paper.pdf.

[18] M. Ito, C.D. Gilbert, Attention modulates contextual influences in the primary visual cortex of alert monkeys, Neuron 5 (1999) 496–498.

[19] M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III, Deep unordered composition rivals syntactic methods for text classification, Proceedings of ACL (2015) 1681–1691.

[20] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: Proceedings of the ACL 2019, Association for Computational Linguistics, Florence, Italy, 2019. pp. 3651–3657. https://www.aclweb.org/anthology/P19-1356, doi: 10.18653/v1/P19-1356..

[21] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, Proceedings of ICLR, 2015.

[22] G. Klein, Y. Kim, Y. Deng, J. Senellart, A. Rush, OpenNMT: Open-source toolkit for neural machine translation, in: Proceedings of ACL 2017, System Demonstrations, Vancouver, Canada, 2017, pp. 67–72.

[23] P. Koehn, Statistical significance tests for machine translation evaluation, in: Proceedings of EMNLP 2004, Barcelona, Spain, 2004. pp. 388–395. https://www.aclweb.org/anthology/W04-3250..

[24] S.M. Kosslyn, W.L. Thompson, I.J. Klm, N.M. Alpert, Topographical representations of mental images in primary visual cortex, Natural 5 (1995) 496–498.

[25] G. Lample, A. Conneau, Cross-lingual language model pretraining, in: the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), 2019..

[26] G. Lample, A. Conneau, L. Denoyer, M. Ranzato, Unsupervised machine translation using monolingual corpora only, in: Proceedings of ICLR, 2018..

[27] I. Lelekas, N. Tomen, S.L. Pintea, Top-down networks: A coarse-to-fine reimagination of cnns, in: arXiv:2004.07629v1, 2020..

[28] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. arXiv preprint arXiv:1910.13461..

[29] S. Maruf, A.F.T. Martins, G. Haffari, Selective attention for context-aware neural machine translation, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3092–3102, https://doi.org/10.18653/v1/N19-1313.

[30] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems (2013) 3111–3119.

[31] S. Mittal, A. Lamb, A. Goyal, V. Voleti, M. Shanahan, G. Lajoie, M. Mozer, Y. Bengio, Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules, in: Proceedings of ICLR, 2020..

[32] K. Papineni, S. Roukos, T. Ward, W.J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of ACL 2002, Philadelphia, Pennsylvania, USA, 2002. pp. 311–318. https://www.aclweb.org/anthology/P02-1040, doi: 10.3115/1073083.1073135..

[33] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, Proceedings of EMNLP 2014 (2014) 1532–1543.

[34] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of NAACL 2018, New Orleans, Louisiana, 2018. pp. 2227–2237. https://www.aclweb.org/anthology/N18-1202, doi: 10.18653/v1/N18-1202..

[35] A. Poliak, Y. Belinkov, J. Glass, B. Van Durme, On the evaluation of semantic phenomena in neural machine translation using natural language inference, in: Proceedings of NAACL 2018, New Orleans, Louisiana, 2018. pp. 513–523. https://www.aclweb.org/anthology/N18-2082, doi: 10.18653/v1/N18-2082..

[36] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, 2018..

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (2019) 9.

[38] A. Raganato, J. Tiedemann, An analysis of encoder representations in transformer-based machine translation, in: Proceedings of EMNLP 2018, Brussels, Belgium, 2018. pp. 287–297. https://www.aclweb.org/anthology/W18-5431, doi: 10.18653/v1/W18-5431..

[39] V. Ramanishka, A. Das, J. Zhang, K. Saenko, Top-down visual saliency guided by captions, Proceedings of CVPR 2017 (2017) 7206–7215.

[40] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of ACL 2016, Berlin, Germany, 2016. pp. 1715–1725. https://www.aclweb.org/anthology/P16-1162, doi: 10.18653/v1/P16-1162..

[41] X. Shi, H. Huang, P. Jian, Y.K. Tang, Improving neural machine translation with sentence alignment learning, Neurocomputing 420 (2020) 15–26, https://doi.org/10.1016/j.neucom.2020.05.104, http://www.sciencedirect.com/science/article/pii/S0925231220313473.

[42] X. Shi, I. Padhi, K. Knight, Does string-based neural MT learn source syntax?, in: Proceedings of EMNLP 2016, Austin, Texas, 2016. pp. 1526–1534. https://www.aclweb.org/anthology/D16-1159, doi: 10.18653/v1/D16-1159..

[43] K. Song, X. Tan, T. Qin, J. Lu, T.Y. Liu, Mass: Masked sequence to sequence pre-training for language generation, International Conference on Machine Learning (2019) 5926–5936.

[44] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Proceedings of NIPS (2014) 3104–3112.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.U. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of NIPS, 2017, pp. 5998–6008..

[46] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, Proceedings of ICML (2008) 1096–1103.

[47] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D.F. Wong, L.S. Chao, Learning deep transformer models for machine translation, in: Proceedings of ACL 2019, 2019a. https://www.aclweb.org/anthology/P19-1176, doi: 10.18653/v1/P19-1176..

[48] Q. Wang, F. Li, T. Xiao, Y. Li, Y. Li, J. Zhu, Multi-layer representation fusion for neural machine translation, Proceedings of COLING (2018) 3015–3026.

[49] X. Wang, Z. Tu, L. Wang, S. Shi, Exploiting sentential context for neural machine translation, in: Proceedings of ACL 2019, Florence, Italy, 2019b. pp. 6197–6203. https://www.aclweb.org/anthology/P19-1624, doi: 10.18653/v1/P19-1624..

[50] C. Xu, C. Paris, S. Nepal, R. Sparks, Cross-target stance classification with self-attention networks, in: Proceedings of ACL 2018, Melbourne, Australia, 2018. pp. 778–783. URL https://www.aclweb.org/anthology/P18-2123, doi: 10.18653/v1/P18-2123..

[51] B. Yang, J. Li, D. Wong, L.S. Chao, X. Wang, Z. Tu, Context-aware self-attention networks, in: Proceedings of AAAI, 2019..

[52] B. Yang, D.F. Wong, T. Xiao, L.S. Chao, J. Zhu, Towards bidirectional hierarchical representations for attention-based neural machine translation, in: Proceedings of the EMNLP 2017, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1432–1441.

[53] M. Yang, R. Wang, K. Chen, X. Wang, T. Zhao, M. Zhang, A novel sentence-level agreement architecture for neural machine translation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020) 2585–2597, https://doi.org/10.1109/TASLP.2020.3021347.

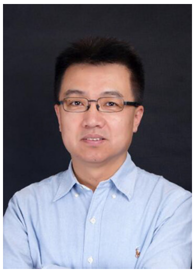
[54] M. Zhong, P. Liu, D. Wang, X. Qiu, X. Huang, Searching for effective neural extractive summarization: What works and what's next, in: Proceedings of ACL 2019, Florence, Italy, 2019. pp. 1049–1058. https://www.aclweb.org/anthology/P19-1100..

[55] Y. Li, J. Li, M. Zhang, Y. Li, P. Zou, Improving Neural Machine Translation with Linear Interpolation of a Short-Path Unit, ACM Transactions on Asian and Low-Resource Language Information Processing 19 (3) (2020).

[56] D. Xu, J. Li, M. Zhu, M. Zhang, G. Zhou, Improving AMR Parsing with Sequence-to-Sequence Pre-training, in: Gao Yang (Ed.), Proceedings of the EMNLP 2020, Association for Computational Linguistics, 2020, pp. 2501–2511.

Yachao Li received the B.S. degree in computer science from the Zhengzhou University, Zhengzhou, China, in 2010, and the M.S. degree in computer science from the Northwest Minzu University, Lanzhou, China, in 2013. He has been working toward the Ph.D degre with the Soochow University, Suzhou, China, from 2016. He also has been a lecturer with the Northwest Minzu University, Lanzhou, China, since 2013. His research interests include machine translation and natural language processing.

Junhui Li received the Ph.D. degree in computer science from Soochow University, Suzhou, China, in 2010. He is currently an associate professor with the School of Computer Science and Technology, Soochow University, Suzhou, China. His current research interests include machine translation, natural language processing. He has published over 20 research papers in reputed journals and conferences, such as AAAI, IJCAI, ACL, EMNLP COLING and TALIIP.



Min Zhang (Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1991 and 1997, respectively. He is currently a Distinguished Professor with the School of Computer Science and Technology, Soochow University, Suzhou, China. His current research interests include machine translation, natural language processing, and artificial intelligence. He has authored 150 papers in leading journals and conferences. He is the Vice President of COLIPS, a Steering Committee Member of PACLIC, an Executive Member of AFNLP and a member of ACL.