Machine Translation (2021) 35:571–593
https://doi.org/10.1007/s10590-021-09282-0

Dual contextual module for neural machine translation

Isaac Kojo Essel Ampomah[1] · Sally McClean[1] · Glenn Hawe[1]

Received: 12 July 2020 / Accepted: 25 September 2021 / Published online: 12 October 2021

Abstract
Self-attention-based encoder-decoder frameworks have drawn increasing attention in recent years. The self-attention mechanism generates contextual representations by attending to all tokens in the sentence. Despite improvements in performance, recent research argues that the self-attention mechanism tends to concentrate more on the global context with less emphasis on the contextual information available within the local neighbourhood of tokens. This work presents the *Dual Contextual* (DC) module, an extension of the conventional self-attention unit, to effectively leverage both the local and global contextual information. The goal is to further improve the sentence representation ability of the encoder and decoder subnetworks, thus enhancing the overall performance of the translation model. Experimental results on WMT'14 English-German (En→De) and eight IWSLT translation tasks show that the *DC* module can further improve the translation performance of the Transformer model.

Keywords Deep neural representation learning · Self-attention networks · Local contexts · Global contexts

1 Introduction

Self-Attention Networks (SAN) (Parikh et al. 2016; Lin et al. 2017) have been shown to significantly improve the performance of neural models for natural language processing (NLP) tasks including Neural Machine Translation (NMT) (Gehring et al. 2017; Vaswani et al. 2017; Dou et al. 2019), acoustic modeling (Sperber et al. 2018), document summarization (Al-Sabahi et al. 2018; Wang and Ren 2018) and reading comprehension (Yu et al. 2018). The state-of-the-art NMT

✉ Isaac Kojo Essel Ampomah
Ampomah-i@ulster.ac.uk

Sally McClean
si.mcclean@ulster.ac.uk

Glenn Hawe
gi.hawe@ulster.ac.uk

[1] Faculty of Computing, Engineering and Built Environment, Ulster University, Belfast, UK

model, the Transformer network (Vaswani et al. 2017), is a prominent non-RNN model based entirely on self-attention. One of the strengths of SANs is their ability to capture long-range dependencies between tokens within a given sentence. SANs achieve this by attending to all tokens present irrespective of their distance apart (Sperber et al. 2018; Yang et al. 2019a). That is, to generate the contextual representation for a token, self-attention tends to consider all tokens in the sentence. The implication of this approach is there is a higher probability that the self-attention mechanism may overlook important short-range dependencies between neighbouring tokens (Sperber et al. 2018; Yang et al. 2019a).

Recently, there has been a growing amount of research dedicated to enhancing the performance of SANs at capturing both the long- and short-range dependencies between the sentence tokens. For example, Sperber et al. (2018) applied a locality restriction approach to limit the attention scope/span of the attention mechanism to the neighbouring elements to further enhance performance on the task of acoustic modeling. Similarly, Yang et al. (2018) employed a learnable Gaussian bias to model localness by revising the attention weight distribution to focus more on a dynamically varying window of tokens. In a different direction, other work (Wu et al. 2019; Yang et al. 2019b) explored convolutional concepts to restricting the attention span to a fixed size window. Even though these approaches enhance the sentence representation ability of the SAN, Xu et al. (2019) argue that restricting the attention scope to some extent limits the ability of the self-attention mechanism to efficiently learn global contextual information.

An interesting question here is the following: how can we effectively exploit local contextual information together with global contextual information without restricting the attention scope of the self-attention mechanism? To this end, this work proposes the *Dual Contextual* (DC) module, an extension to the self-attention unit to leverage both the local and global contextual information to further enhance translation quality. The *DC* module shown in Fig. 1 comprises two sub-modules, namely the *Local Contextual Unit* and the *Context Interaction Unit*. The Local Contextual Unit is a CNN-based network employed to capture the local contextual information respective to a neighbourhood window determined by the convolution filter size. The generated sentence representation from the Local Contextual Unit is passed to the Context Interaction Unit which employs two multi-head attention-based units and an aggregation unit to model the interaction between the global and local contextual information. The proposed approach imposes no restrictions on the attention scope, allowing the self-attention to fully capture any long-range dependencies. The learning of the short-range dependency information is assigned to the Local Contextual Unit. One of the multi-head attention units in the Context Interaction Unit is employed to perform the self-attention computation across the input representation. The *DC* module is incorporated into the Transformer network. Experimental results on nine translation tasks show that the proposed *DC* module can further improve the translation quality of the Transformer model. Furthermore, analysis based on ten linguistic probing tasks (Conneau et al. 2018) demonstrates that the leveraging both global and local contextual information further enhances the model's ability to effectively capture the necessary linguistic properties required for the source translation task.

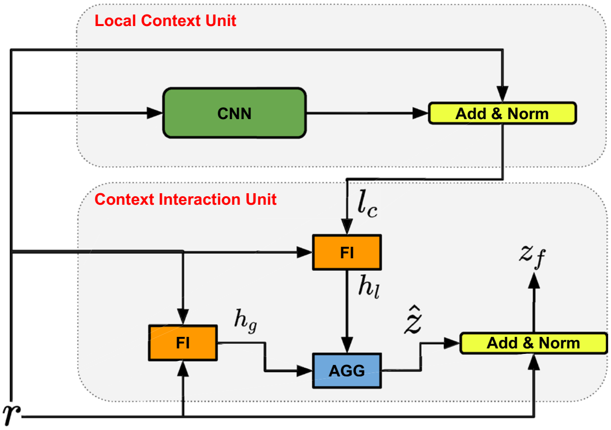


Fig. 1 The proposed *DC* module employed to leverage both the local and global information. z_f is the context-rich representation generated based on the input sentence representation r. l_c is the contextual representation generated by the *Local Contextual Unit* and *AGG* denotes the aggregation unit employed to combine the hidden representations h_l and h_g generated by *Feature Interaction* (FI) units

The contributions of this work are the following:

1. Proposing the *Dual Contextual* module to leverage both the local and global contextual information to further improve translation performance.
2. Demonstrating consistent improvement over the strong Transformer baselines across nine translation tasks.
3. Providing analysis on the performance impact of limiting the application of the *DC* module to the layers of either the encoder subnetwork or the decoding subnetwork.
4. Providing an ablation study on the impact of limiting the contextual modeling via the *DC* module to only some combinations of encoding layers.

The remainder of the paper is organized as follows: Sect. 2 briefly reviews the related literature and Sect. 3 provides a background to NMT. The proposed *Dual Contextual* module is presented in Sect. 4. The experiments conducted are presented in Sect. 5, and the results are compared and discussed in Sect. 6. Section 7 presents a detailed analysis on the impact of the *DC* module on translation performance of the Transformer network. Finally, the conclusions are presented in Sect. 8, together with avenues for future work.

2 Related work

Leveraging useful word-to-word contextual information has been shown to be essential to achieve higher translation quality in statistical machine translation (SMT) models (Gimpel and Smith 2008; He et al. 2008; Marton and Resnik 2008). One of the earliest works exploiting both local and global contextual information to improve NMT performance was Luong et al. (2015). Specifically, to utilize both local and global information of the source tokens, the authors employ two attention mechanisms, namely the global and local attention units. During the decoding step *t*, the global attention considers all the source tokens, whereas the local attention computation is performed across a subset of the source tokens. The improvements in performance highlight the importance of capturing both global and local contextual information. Motivated by this, recent work (Sperber et al. 2018; Yang et al. 2018; Xu et al. 2019) has investigated strategies to learn both long-distance and short-range dependencies between tokens to further enhance the sentence representational ability of the self-attention mechanism. To model localness, Yang et al. (2018) proposed a modification of the self-attention mechanism with a learnable Gaussian bias which specifies the central position and window of tokens neighbourhood to which more attention should be paid. Similarly, Sperber et al. (2018) explored two masking techniques for controlling the contextual range of self-attention for the task of acoustic modeling. Other work (Wu et al. 2019; Yang et al. 2019b) has explored convolutional concepts to restrict the attention scope/span to a window of neighbouring tokens.

The ability of self-attention to efficiently capture global contextual information has been identified as one of its salient strengths, improving the performance of downstream NLP tasks such as semantic modeling (Yang et al. 2019a) and constituency parsing (Kitaev and Klein 2018). However, the approaches by Wu et al. (2019) and Yang et al. (2019b) restricting the attention scope to some degree can result in loss of important global and long-distance dependencies (Xu et al. 2019). The work of Song et al. (2018b) and Xu et al. (2019) exploit a hybrid attention mechanism to learn local and global contextual information. These approaches augment the self-attention mechanism with masking designed purposely to learn the local contextual patterns without restricting the self-attention mechanism's ability to efficiently model the global and long-distance dependencies. Another form of hybrid attention mechanism is the *QANet* proposed by Yu et al. (2018), which employs a multi-layer depth-wise separable CNN to initially model the local contextual representation before the application of the self-attention unit. The resulting model significantly outperformed RNN-based models for the task of machine comprehension. In a similar direction, our work proposes the *Dual Contextual Module* to leverage both local and global information to further improve the performance of NMT. However, unlike Shen et al. (2018); Song et al. (2018b); Yu et al. (2018) and Xu et al. (2019), the *DC* module employs two separate attention computation units. Furthermore, the modeling of the local and global contextual information is applied to all layers within the encoder and decoder subnetworks. This work argues that the decoder subnetwork requires rich contextual source representation to achieve higher translation quality. This implies that the output of the encoder subnetwork has to 'fully' encapsulate both the local and global contextual information of the source tokens.

3 Background

A typical NMT system learns the conditional probability $\mathsf{P}(Y \mid X;\theta)$ mapping from a given source sequence $X = \left(x_1, x_2, \cdots, x_M\right)$ to a target language sentence $Y = \left(y_1, y_2, \cdots, y_N\right)$ of length N, where x_i and y_t denote the i^{th} and t^{th} tokens of X and Y respectively. The model parameter θ is learned by maximizing the likelihood, as in (1):

$$\mathsf{P}(Y \mid X;\theta) = \prod_{t=1}^{M} \mathsf{P}\left(y_t \mid y_{<t}, X;\theta\right) \tag{1}$$

where $y_{<t} = y_1, \cdots, y_{t-1}$ is the partial target sequence generated.

NMT models usually consist of two subnetworks, namely the *Encoder* and *Decoder*. The Encoder generates the hidden representation $H^e = [h_1^e, h_2^e, \cdots, h_M^e]$ based on the source embedding $E_x = [e_1, e_2, \cdots, e_M]$, where $e_i \in \mathbb{R}^d$ is the embedding vector for the i^{th} source token (i.e. x_i). During the decoding step t, the target token y_t is generated by the decoder subnetwork based on the partial target sequence $y_{<t}$ and source semantic representation H^e from the encoding subnetwork.

3.1 The transformer model

Similar to NMT models based on RNNs (LSTM/GRU) (Cho et al. 2014; Bahdanau et al. 2015) and CNN (Gehring et al. 2017), the Transformer network (Vaswani et al. 2017) employs the Encoder-Decoder architectural structure. The encoder and decoder subnetworks consist of a stack of L-identical layers. For the encoding subnetwork, each layer consists of two sublayers, a self-attention (SA) module followed by a position-wise feed-forward network. A decoding layer is composed of a similar network structure to that of an encoding layer, but it employs an encoder-decoder attention unit in between the SA module and position-wise feed-forward network. To enhance the flow of gradient information, both the encoder and decoder subnetworks employ residual connection (He et al. 2016) and layer normalization (Ba et al. 2019) around each sublayer.

A linear transformation layer with a softmax activation is employed to convert the decoder's output representations H_d^L into output probabilities over the target vocabulary. To further improve the model's performance, recent work (Inan et al. 2016; Pappas et al. 2018) has proposed a linear transformation layer sharing the same weights with the word embedding layers of the decoder and encoder subnetworks. Furthermore, this strategy reduces the size of the model in terms of the number of trainable parameters.

3.2 Self-attention mechanism

An attention mechanism aims at modelling the direct relations between tokens of a given pair of sequence representations. Formally, the attention mechanism generates the token-to-token contextual representation c in (2):

$$\begin{aligned} c &= \mathrm{ATT}(Q, K, V) \\ \mathrm{ATT}(Q, K, V) &= \alpha \cdot V \\ \alpha &= \operatorname{softmax}(\operatorname{score}(Q, K)) \end{aligned} \tag{2}$$

where $Q \in \mathbb{R}^{Z \times d_{model}}$, $K \in \mathbb{R}^{J \times d_{model}}$, and $V \in \mathbb{R}^{J \times d_{model}}$ are the query, key and value vectors, respectively. α is the attention weight distribution computed across the tokens represented by K. Z and J are the number of tokens in the given sequence representations Q and K respectively. d_{model} denotes dimension of the hidden representation. Finally, score $(\cdot)$ is a scaled dot product function defined as in (3):

$$\operatorname{score}(Q, K) = \frac{Q \times K^{\mathsf{T}}}{\sqrt{d_k}} \tag{3}$$

where $\sqrt{d_k}$ is a scaling factor employed to stabilise the attention computation.

Self-attention, a variant of the attention mechanism, performs the attention computation across a single sentence representation. Specifically, self-attention models the long-range dependencies between the tokens within the input sequence. Recent work including Gehring et al. (2017); Vaswani et al. (2017); Shaw et al. (2018) and Yu et al. (2018) has shown the potential performance gain of the self-attention mechanism over RNN by capturing long-range contextual information and dependencies between the pairs of tokens within the given sequence.

For a layer[1] l, the self-attention unit computes the sentence representation h^l by attending to the hidden representation H^{l-1} from the preceding layer[2] $(l - 1)$. The first stage of the self-attention unit is the computation of the query (Q), value (V) and key (K) based on three separate projections of the H^{l-1}, as in (4):

$$\begin{aligned} Q &= H^{l-1} W^{Q} \in \mathbb{R}^{J \times d_{model}} \\ V &= H^{l-1} W^{V} \in \mathbb{R}^{J \times d_{model}} \\ K &= H^{l-1} W^{K} \in \mathbb{R}^{J \times d_{model}} \end{aligned} \tag{4}$$

where $\{W^V, W^Q, W^K\} \in \mathbb{R}^{d_{model} \times d_{model}}$ are the projection weights employed to generate the value, query and key vectors respectively. The self-attention unit employed by the Transformer model is based on the multi-head attention (MHA) mechanism. Specifically, the MHA defines N_h attention heads where each $head_i$ generates a separate attention weight distribution α^i. The attention head $head_i$ attends to tokens at different positions based on the α^i when generating the contextual representation $c^i \in \mathbb{R}^{J \times \frac{d_{model}}{N_h}}$. The MHA is formulated as in (5):

[1] A layer of either the encoder or decoding subnetwork.

[2] For $l = 0$, H^{l-1} is the output of the embedding layer.

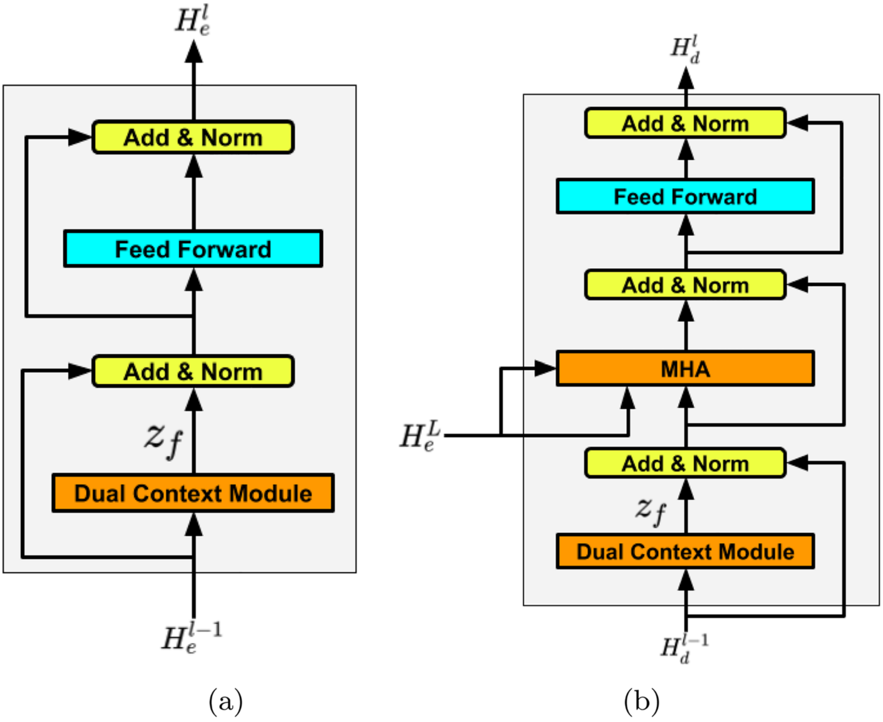


Fig. 2 Illustration of the application of the proposed *DC* module to (**a**) the encoder layer and (**b**) the decoder layer. H_d^{l-1} and H_d^l denote the output representations from the decoding layers $l-1$ and l, respectively. Similarly, H_e^{l-1} and H_e^l are the input and output representations of the encoder layer l. H_e^L is the output of the top-level encoding layer passed to the encoder-decoder MHA sublayer of the decoding layer

$$
\begin{aligned}
h^l &= \mathrm{MHA}(Q, K, V) \\
\mathrm{MHA}(Q, K, V) &= O_c W_o \\
O_c &= \mathrm{Concat}\,(c^1, c^2, \cdots, c^{N_h}) \\
c^i &= \mathrm{ATT}(Q^i, K^i, V^i)
\end{aligned}
\tag{5}
$$

where $W_o \in \mathbb{R}^{d_{model} \times d_{model}}$ is a trainable weight parameter employed to generate the output representation $h^l \in \mathbb{R}^{J \times d_{model}}$ based on concatenation of the contextual representations from all the attention heads O_c. $\{Q^i, K^i, V^i\} \in \mathbb{R}^{J \times d_{\mathrm{mod}\,el}/N_h}$ are the query, key and value vectors with respect to the i^{th} attention head, respectively.

4 Dual contextual module

Our approach seeks to further improve translation performance by leveraging both the local and global contextual information to enhance the sentence representation ability of the encoding and decoding subnetworks. To this end, for a given layer (of either encoder or decoder subnetwork), the self-attention unit is replaced with a

Dual Contextual Module as illustrated in Fig. 2. As shown in Fig. 1, the *DC* module consists of two sub-modules, namely the Local Contextual Unit and the Context Interaction Unit. To enhance the flow of gradient information, residual connections and layer normalization are applied across the output of each sub-module. For an encoder layer l, the input sentence representation to the *DC* module (r) denotes the output of the $(l-1)^{th}$ encoder layer H_e^{l-1}. Similarly, for a decoding layer l, r is the output of the preceding layer $l-1$ (i.e. H_d^{l-1}).

4.1 Local context unit

To capture local contextual information, this unit employs a single layer one-dimensional (1-D) convolution network with Gated Linear Unit (GLU) activation (Dauphin et al. 2017). The 1-D convolution can learn local dependencies between the source tokens within a neighbourhood of width determined by the kernel size of the filter (Song et al. 2018a; Yu et al. 2018). The local contextual representation l_c is generated as in (6):

$$\begin{aligned} l_c &= \text{LayerNorm}\,(\hat{r} + r) \\ \hat{r} &= \text{Concat}\,(\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_J) \\ \hat{r}_t &= \text{g}\big([r_{t-f/2}, \cdots, r_{t+f/2}]W^r + b^r\big) \end{aligned} \tag{6}$$

where $W^r \in \mathbb{R}^{f \cdot d_{model} \times 2 \cdot d_{model}}$ and f are the convolution filter and kernel size, respectively. b^r is the bias and $\text{g}(\cdot)$ the GLU activation. $\hat{r}_t$ is the local contextual representation of the tth token in the sequence. As shown in (6), $\hat{r}$ is generated from the concatenation of the local contextual representations with respect to all input tokens.

4.2 Context interaction unit

This is the core of the *DC* module computing the context-rich representation z_f based on the l_c from the Local Contextual Unit and r. Given l_c and r, the Context Interaction Unit generates the hidden representations h_l and h_g via two *Feature Interaction* (FI) units as shown in Fig. 1. Each FI employs multi-head attention mechanism to model the interactions between the unit's inputs, as in (7):

$$\begin{aligned} h_l &= \text{FI}(r, l_c) \\ h_g &= \text{FI}(r, r) \\ &\text{where} \\ \text{FI}(A, B) &= \text{concat}(M_1, M_2, \cdots, M_{n_h}) \\ M_i &= \text{ATT}(AW_i^q, BW_i^k, BW_i^v) \end{aligned} \tag{7}$$

where A and B represent the inputs to the FI units. $AW_i^q, BW_i^k, BW_i^v \in \mathbb{R}^{J \times \frac{d_{model}}{N_h}}$ are the projections of the query (A), key (B) and value (B) vectors with respect to the attention head $head_i$ sub-space, respectively. As shown in Fig. 1, h_g is the global contextual representation obtained from a self-attention operation across the r. The

h_l is sentence representation generated from the attention operation between r and the l_c (from the Local Contextual Unit).

A feature *Aggregation* unit (AGG) is employed to generate the hidden representation $\hat{z}$ from the combination the outputs of the FI units, as in (8):

$$\hat{z} = [h_l;h_g]W_z + b_z \tag{8}$$

where $W_z \in \mathbb{R}^{2\cdot d_{model} \times d_{model}}$ and $b_z \in \mathbb{R}^{d_{model}}$ are trainable model parameters. $[\cdot;\cdot]$ denotes the concatenation operation. Formally, the Context Interaction Unit computes the z_f as in (9):

$$z_f = \text{LayerNorm}\,(\hat{z} + r) \tag{9}$$

As shown in Fig. 2, the generated $z_f \in \mathbb{R}^{J\times d_{model}}$ is passed to the subsequent sublayers for further processing. For the encoder subnetwork, it is fed into the position-wise feed-forward network. However, in the case of the decoder subnetwork, it is one of the inputs to the encoder-decoder attention unit.

5 Experimental setup

5.1 Datasets

The effectiveness of the proposed approach is evaluated on WMT'14 English-German (En→De), and IWSLT tasks: French-English (Fr↔En), Spanish-English (Es↔ En), Romanian-English (Ro↔En) and English-Vietnamese (En↔Vi) translation tasks.

The dataset for the En→De translation task consists of about 4.5M sentence pairs for training. Consistent with existing works (Vaswani et al. 2017; Yang et al. 2018; Xu et al. 2019), the newstest2013 and newstest2014 sets are employed as the validation and test sets, respectively. For the Es↔En and Ro↔En tasks, the datasets employed are respectively from the Spanish-to-English and Romanian-to-English translation tracks of the IWSLT 2014 evaluation campaign (Cettolo et al. 2014).[3] The training sets for these tasks consist of 183k and 182k sentence pairs, respectively. The test set is the tst2014 split and the validation set is created by concatenating the tst2010, tst2011, tst2012 and dev2010 sets. The En↔Vi translation task is performed on the English to Vietnamese track of IWSLT 2015 (Cettolo et al. 2015). The dataset consists of 133k training sentence pairs. The tst2013 (consisting of about 1.2k sentences pairs) and tst2012 (consisting of 1553 sentences pairs) are employed as the test and validation data, respectively. For the Fr↔En task, the dataset consisting of 207k training sentence pairs is from the IWSLT 2015 campaign. The test set is from the combination of the tst2014 and tst2015 splits. The validation set is the tst2013 set.

[3] https://wit3.fbk.eu/mt.php?release=2014-01.

For a given translation task, the size of the vocabulary is limited to a few numbers of words to reduce the complexity of the model. However, limiting the vocabulary size usually results in the out-of-vocabulary problem, which can be mitigated by learning a shared vocabulary via byte-pair-encoding[4] (Sennrich et al. 2016) for both the encoding and decoding subnetworks. The resulting subword-based shared vocabulary is employed to encode the sentence pairs (source and target sentences).[5] The vocabularies for the En→De, Es↔En, En↔Vi, Ro↔En and Fr↔En translation tasks consist of 32k, 34k, 21k, 32k and 31k subword tokens, respectively.

The translation quality of the proposed model is reported based on the BLEU metric (Papineni et al. 2002). Specifically, the 4-gram case-sensitive NIST BLEU metric is employed as the evaluation metric for the En→De task. The translation quality for En↔Vi is reported based on the case-sensitive BLEU score computed with sacreBLEU.[6] Finally, for the other IWSLT translation tasks, the case-sensitive BLEU metric evaluated with multi-bleu.pl[7] is employed. The statistical significance of the BLEU scores between the baseline and our *DC* based models is evaluated with paired bootstrap resampling (Koehn 2004) using the compare-mt[8] (Neubig et al. 2019) library with 1000 resamples. For simplicity, the statistical significance test is performed on the WMT'14 En→De task mainly due to the size of the test dataset.

5.2 Model settings

The proposed *Dual Contextual Module* is integrated into the Transformer network (Vaswani et al. 2017). The parameter settings such as the number of encoder and decoder layers, number of attention heads and hidden size employed for each translation task are chosen based on the number of sentence-pairs within the training set. For the low-resource (IWSLT) tasks, the model consists of 4-layer encoder and 4-layer decoder subnetworks each with hidden size and number of attention heads set as 256 and 4, respectively. However, for En→De, the hidden size, number of attention heads and number of layers are 512, 8 and, 6, respectively. Based on an analysis performed on the En→De task (see Sect. 7.2), the filter size of the CNN unit of the *Local Context Unit* is set to 2.

5.3 Training and inference

For the IWSLT tasks, the models are trained with a batch size of 2048 tokens and the number of training iterations is 200k. The batch size and the number of iterations employed to train the model on the En→De are 4960 tokens and 160k iterations, respectively. Adam (Kingma and Ba 2014) (with

[4] https://github.com/rsennrich/subword-nmt.

[5] The original casing for the tokens in each sentence is preserved.

[6] https://github.com/awslabs/sockeye/tree/master/contrib/sacrebleu.

[7] https://github.com/moses-smt/mose.sdecoder/blob/master/scripts/generic/multi-bleu.perl.

[8] https://github.com/neulab/compare-mt.

Table 1 Translation performance on the WMT14 English-German (En→De) test set

Model	# Params	Train	BLEU
8-Layer RNN (Wu et al. 2016)	–	–	26.30
ConvSeq2Seq (Gehring et al. 2017)	–	–	26.36
Transformer-Base (Vaswani et al. 2017)	–	–	27.31
Transformer+CNN (Yu et al. 2018)	–	–	27.70
Transformer+EM Routing (Dou et al. 2019)	–	–	28.81
Transformer+Layer Aggregation (Dou et al. 2018)	–	–	28.78
Transformer+2D-CSANs (Yang et al. 2019b)	–	–	28.18
Our NMT Systems			
Transformer	61.2M	3.66	28.37
Enc-DC	73.86M	3.0	29.26 (+0.89)‡
Dec-DC	73.86M	3.08	28.86 (+0.49)‡
Full-DC	86.5M	2.53	29.11 (+0.74)‡

The progressive gain between our implementation of the Transformer baseline and our approach is shown in parentheses

#**Params** denotes the number of trainable parameters per model. **Train** indicates the training speed (steps/second)

‡indicates statistically significant difference with $\rho < 0.01$

Table 2 Translation performance on the IWSLT tasks ({Vi, Es, Fr, Ro }↔ En)

Task	Models			
	Transformer	Enc-DC	Dec-DC	Full-DC
Vi→En	29.03	29.42	29.20	**29.70**
En→Vi	30.58	**31.28**	31.24	31.10
Es→En	39.80	**40.18**	39.96	40.15
En→Es	37.90	**38.59**	38.28	38.10
Fr→En	32.85	33.35	33.15	**33.42**
En→Fr	33.15	**33.94**	33.43	33.60
Ro→En	26.66	**27.15**	26.81	27.14
En→Ro	20.91	21.20	21.22	**21.27**

$\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^9$) is used as the optimizer to train the models. Unlike Vaswani et al. (2017), the learning rate scheduling algorithm employed in this work is the *single-cosine-cycle* with warm-up (So et al. 2019).

During inference, the target translations are generated using the beam search algorithm. For the En→De task, a beam-size of 4 and a length penalty of 0.6 is employed. Finally, for the IWSLT tasks, the beam size and length penalty are 6 and 1.1, respectively. The proposed *DC* module can be applied to both subnetworks of the Transformer model. Furthermore, it can also be limited to either the encoder or decoder subnetwork. For simplicity, the Transformer model trained

Table 3 Existing results on the IWSLT En→Vi and Es→En translation tasks

Model	BLEU
En→Vi	
Luong & Manning (Luong and Manning 2015)	23.30
NPMT (Huang et al. 2018)	27.69
NPMT + LM (Huang et al. 2018)	28.07
Es→En	
UEDIN (Cettolo et al. 2014)	37.29
Tied Transformer (Xia et al. 2019)	40.51
Layer-wise Coordination (He et al. 2018)	40.50

with the *DC* module applied to both the encoder and decoder subnetworks is denoted as Full-DC. The model with the *DC* module applied to only the encoder is denoted as Enc-DC. Finally, Dec-DC is the Transformer model trained with the *DC* module employed within only the decoder subnetwork.

6 Results

This section presents the evaluation performance of the proposed *Dual Contextual* model on the translation tasks under consideration. Table 1 shows translation performance on the WMT'14 En→De dataset, with the value in parentheses denoting the progressive gain over the Transformer baseline. Similarly for the IWSLT tasks, the results are summarized in Table 2. Finally, Table 3 summarizes the performance of existing models on the IWSLT En→Vi and Es→En translation tasks. On the En→De translation task, the *DC* unit significantly improves the performance of the Transformer model by +0.89 BLEU in the case of the Enc-DC model and +0.49 BLEU with respect to the Dec-DC model. The performance of the Dec-DC model is improved by +0.25 BLEU when the *DC* module is applied to both subnetworks. However, the performance achieved by the Full-DC model is lower than that achieved by the Enc-DC model. Besides, the translation performance obtained via the *DCM* unit is higher than all the existing models further demonstrating the superiority of the proposed approach.

The languages under consideration for the IWSLT tasks belong to different language families. The overall performance of an NMT model is affected by the linguistic and syntactic properties or structures of the language pairs under consideration. The translation quality achieved demonstrates the effectiveness of our proposed *DC*-based models at translating between languages of different families as shown in Table 2. On average, the Enc-DC model consistently achieves a higher performance gain over the Transformer baseline across all the IWSLT translation tasks. Compared to existing work, the baseline Transformer model consistently outperforms the models of Luong and Manning (2015) and Huang et al. (2018), with a performance improvement of +2.51 BLEU score on the En→Vi translation task as summarized in Tables 2 and 3. The proposed Enc-DC and Dec-DC models achieved BLEU scores of 31.28 and 31.24, respectively. This represents a further performance gain up to +0.7. In contrast, a lower gain in translation quality of +0.52 BLEU is achieved when *DC* is applied to both subnetworks (i.e. Full-DC). Similarly, on the Es→En task, all our models outperform the Transformer baseline. A marginal performance gain of +0.16 BLEU is achieved when the *DC* module is applied to only the decoder subnetwork (i.e. Dec-DC). On this dataset, the best performance is achieved by the Enc-DC and Full-DC models enhancing the translation quality by +0.38 BLEU and +0.35 BLEU, respectively. However, the translation performance of the Enc-DC, Dec-DC and Full-DC models is lower than that achieved by the models in He et al. (2018) and Xia et al. (2019), outperforming only the model of Cettolo et al. (2014).

Overall, the performance achieved across all translation tasks indicates the benefits of leveraging both the local and global contextual information. The Enc-DC model consistently outperforms the Dec-DC approach on all datasets. This could be attributed to the bias/mask employed by self-attention units within the decoding layers. During the decoding step t, the attention bias limits the decoder's self-attention to only consider the target sub-sequence (i.e. $y_{<t}$) generated so far. This implies that the decoder layer is able to exploit the local information within the neighbourhood of $y_{<t}$. Therefore unlike the encoder subnetwork, the CNN unit generating l_c has a limited impact on the performance of the decoding layers and the decoder subnetwork. This is consistent with the observations made by Zhang et al. (2018). Furthermore, applying the *DC* module to both subnetworks (Full-DC) in most cases only improves performance from the Dec-DC model's perspective. Besides, the Full-DC achieved a higher performance gain over the Enc-DC model only on the Vi→En task. On a number of the translation tasks under consideration, there is no significant difference in the performance of the Full-DC and Enc-DC models.

As shown in Table 1, the *DC* module introduces new parameters due to the additional attention computation as well as the LC units. Applying the *DC* module to all the subnetworks (Full-DC model) results in about a 25M increase compared to the 12.7M by the Enc-DC and Dec-DC models. The computational speed of a neural model is affected by quantities such as the optimizer, number of network parameters and the other computations that directly modify the formulation of the network (Popel and Bojar 2018). Accordingly, our *DC*-based models have lower training speeds compared to the baseline. Compared to the Full-DC model, the Enc-DC and Dec-DC models are shown to have the least decrease in training speed. Based on the translation quality achieved and the computation speeds, this work suggests limiting the explicit modeling of the local contextual information via the DC module to only the encoding subnetwork.

7 Analysis

This section presents analyses performed to better understand the performance improvement introduced by the *Dual Contextual Module*. These analyses are performed on the En→De dataset. As shown in Sect. 6, the best performance is obtained when the *DC* module is applied across the encoder subnetwork, so the analyses presented in Sects. 7.1, 7.2 and 7.3 are performed only on the Enc-DC model.

7.1 Linguistic probing task

As shown in Tables 1 and 2, the Enc-DC model further improves the performance of the Transformer baseline model across all the translation tasks under consideration. However, little is know about the linguistic perspectives or properties improved by the proposed module. Following Conneau et al. (2018), Li et al. (2019), ten classification tasks are conducted to study the linguistic properties enhanced by the *DC* module. The ten classification tasks are divided into three categories:

Surface (Surf) focuses on the surface properties captured by the sentence representation or embedding. This category consists of the *Sentence Length* (SentLen) and *Word Content* (WC) tasks. Under the SentLen task, the goal is to predict the length of the input sentence based on the number of its tokens. The WC task tests the possibility of recovering information about the original word in a sentence given its embedding.
Syntactic (Sync) evaluates the ability of the encoder subnetwork to learn/capture syntactic information. The syntactic tasks under consideration are: *Tree Depth* (TDep), *Bigram Shift* (BShift) and *Top Constituent* (ToCo). The TDep task tests the capability of the encoder to infer the hierarchical structure of the input sentence. For the BShift, the sensitivity of the encoder to the legal word order is evaluated. Specifically, the goal is to predict if two consecutive tokens within the sentence have been inverted or not. Finally, under the ToCo task, the sentence is classified in terms of the sequence of top constituents immediately below the sentence node.
Semantic (Sem) evaluates the capabilities of the encoder to understand the denotation of a given sentence. To achieve higher performance on this task, the sentence embedding should encapsulate the syntactic structure (Conneau et al. 2018). Here, there are five tasks under consideration. The first semantic probing task is the *Tense* classification task which evaluates the tense of the main clause verb (whether it is in the past or present tense). The next is the *Subject Number* (SubjN) which focuses on predicting the number of the subject in the main clause. In contrast, the *Object Number* (ObjN) predicts the number of the direct object of the main clause. Under the *Semantic odd man out* (SoMo) task, the sentences are modified by randomly replacing a noun or verb with another noun or verb. Here the task is to predict whether a sentence has been modified or not. Finally for the *Coordination Inversion* (CoIn), the sentences are divided into two coordinate clauses. In half of the sentences, the order of the clauses is inverted and the task here is to check whether a sentence is modified or is left intact.

The above tasks are performed based on the sentence representations generated by the encoding subnetwork. Specifically, the decoding subnetwork of the pre-trained

Table 4 Classification performance on the ten probing tasks to evaluate the linguistic information ("Surface", "Syntactic" and "Semantic") learned by the encoding subnetwork of the Transformer baseline and our proposed model

Tasks		Models	
		Transformer	Enc-DC
Surface	SentLen	93.38	93.45
	WC	71.50	75.95
	#Avg	82.44	**84.70**
Syntactic	TDep	43.93	44.29
	BShift	69.49	74.11
	ToCo	74.56	75.44
	#Avg	62.66	**64.61**
Semantic	Tense	88.40	88.65
	SubjN	85.01	85.33
	ObjN	85.9	85.13
	SoMo	52.67	52.49
	CoIn	62.30	62.56
	#Avg	**74.86**	74.83

"#Avg" denotes the average score across the sub-tasks under each category

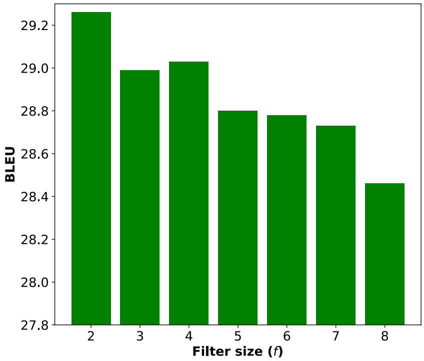
NMT model is replaced with a two-layer classifier. L2 regularization of $\lambda = 1e^{-4}$ is applied to the hidden layer of the classifier. For each classification task, the input representation to the classifier is the mean of the output representation from the top-level (last) encoding layer. The resulting classification models are trained and evaluated on the dataset presented by Conneau et al. (2018).[9] The training corpus for each task comprises 100k sentences, with 10k sentences for validation and 10k sentences for testing. During training, only the parameters of the classifier are updated. The classifiers are trained for 100 epochs with the RMProp optimizer using the learning rate of $2.5e^{-4}$ and mini-batch size of 64. An early stopping criterion is applied during training based on the accuracy score on the validation set. The parameters of the pre-trained encoding subnetwork are fixed to quantify the linguistic properties and information captured by the pre-trained encoder subnetwork of the Transformer baseline and our Enc-DC model.

7.1.1 Results

The results of the probing tasks are summarized in Table 4. Despite the *DC*-based encoding subnetwork of the Enc-DC model achieving best performance on three of the five ***Semantic*** tasks, on average it achieved identical performance as the baseline. The actual performance gain introduced by the *DC* module can be seen across the ***Surface*** and ***Syntactic*** tasks. Across these tasks, the Enc-DC model significantly outperformed the Transformer baseline. Tasks such as WC and CoIn require global contextual information, whereas local contextual information is generally required

[9] https://github.com/facebookresearch/SentEval/ree/master/data/probing.

Fig. 3 Impact of f (the convolution filter size employed by the *Local Context unit*) on the performance of the Enc-DC model

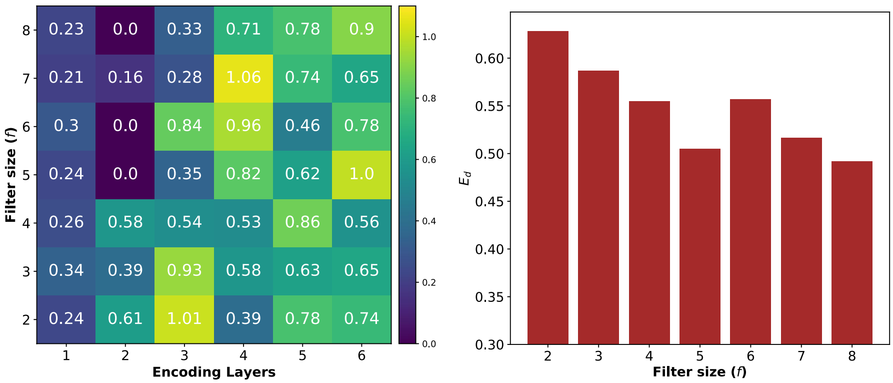


to achieve higher performance on tasks such as ToCo, BShift and SentLen. Overall, the performance on the ***Syntactic***, ***Surface*** and ***Semantic*** probing tasks demonstrates the effectiveness of the *DC* module in learning both the global and local contextual information required to achieve higher translation performance. The *DC* module allows the encoder subnetwork to effectively learn the surface and syntactic information of the source sentence with minimal impact on its ability to encode the deeper linguistic features.

7.2 Effect of CNN kernel size

To analyse the impact of the CNN kernel size (employed by the *Local Context unit*) on translation quality, different Enc-DC models were trained with the kernel size $f \in [2, 3, 4, 5, 6, 7, 8]$. The translation performance of each Enc-DC model is summarized in Fig. 3. As shown, the Enc-DC models achieved almost identical BLEU scores when trained with the filter size $5 \leq f \leq 7$. This is also true for the values of $f \in [3, 4]$. The worse performance is obtained with $f = 8$, producing a marginal performance gain of about +0.10 BLEU over the Transformer baseline. However, the best performance is achieved with $f \in [2, 3, 4]$. Overall, the performance of the Enc-DC model generally decreases as the filter size increases. One possible reason for this is that for larger filter sizes there is a higher possibility of information overlap between l_c and h_g. As a result, the *FI* unit generating the h_l (based on l_c and H_e^{l-1}) has only a limited impact on the performance of the *DC* unit at learning the contextual sentence representation. This is because the generation of the h_g and h_l will be more meaningful if each *FI* unit captures diverse information.

The above hypothesis is investigated by analyzing the relationship between the attention weight distributions α_g and α_l associated with the generation of the hidden contextual representations h_g and h_l, respectively. As mentioned in Sect. 4.2, each *FI* unit is a multi-head attention unit employing n_h attention heads. For simplicity, the attention weights for each encoding layer are represented by the attention head with the maximum weight distribution among all the n_h heads employed by the *FI* units, as in (10):



(a) (b)

Fig. 4 Variation of the divergence between attention weights α_g and α_l for the difference filter size $f \in [2, 3, 4, 5, 6, 7, 8]$. **a** Divergence across each encoding layer with respect to the difference filter sizes. **b** Mean divergence (E_d) across all the layers within the encoding subnetwork for each filter size under consideration

$$\begin{aligned} \alpha_g &= \max\left([\alpha_g^1, \alpha_g^2, \cdots, \alpha_g^{n_h}]\right) \\ \alpha_l &= \max\left([\alpha_l^1, \alpha_l^2, \cdots, \alpha_l^{n_h}]\right) \end{aligned} \tag{10}$$

where α_g^i and α_l^i are the weight distribution employed by the i^{th} attention head, respectively. To this end, Jensen-Shannon divergence, $JS\ P\ Q$, is employed as a distance metric to measure how far the α_g and α_l are from each other. The distance is measured using $JS\ P\ Q$ because it is symmetric and bounded. Given the multi-dimensional attention weight distributions α_l and α_g, the divergence with respect to the encoding layer l, dv^l is computed as in (11):

$$\begin{aligned} dv^l &= JS\alpha_g\alpha_l \\ JS\alpha_g\alpha_l &= \frac{1}{2} \times KL\alpha_g m + \frac{1}{2} \times KL\alpha_l m \\ m &= \frac{1}{2} \times \left(\alpha_g + \alpha_l\right) \end{aligned} \tag{11}$$

where $KL\ P\ Q$ is the Kullback-Leibler divergence formulated as in (12):

$$KLPQ = \sum_x P(x) \log P(x) - \sum_x P(x) \log Q(x) \tag{12}$$

The overall divergence (E_d) for the L-layer encoding subnetwork is computed as the average of the divergence (dv^l) computed for each encoding layer, as in (13):

$$E_d = \frac{1}{L} \sum_{l=1}^{L} dv^l \tag{13}$$

Table 5 Translation performance of different combinations of encoder layers of the Enc-DC model

#	Layers	BLEU	$\varDelta$
1	[1-6]	29.26	-
2	[1-1]	29.04	−0.22
3	[1-2]	28.98	−0.28
4	[1-3]	28.93	−0.33
5	[1-4]	29.04	−0.22
6	[1-5]	28.94	−0.32
7	[5-6]	28.47	−0.79
8	[4-6]	28.57	−0.69
9	[3-6]	28.78	−0.48

[*i*-*j*] denotes limiting the application of the *DC* module to the encoding subnetwork from layer *i* to layer *j*. $\varDelta$ indicates the difference in performance between the [1-6] and the [*i*-*j*] encoder layer combinations

Larger values of E_d indicate that, on average across the multiple encoding layers, the two *FI* units (of the *DC* module) can capture more diverse information using the attention weights α_l and α_g.

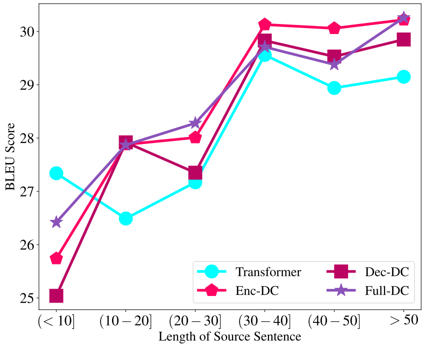
Fig. 4 illustrates the divergence between α_l and α_g for the different filter kernel sizes. As displayed in Fig. 4a, variation in the divergence across the encoding layers is dependent on the value of *f* employed to train the Enc-DC model. In all cases, the divergence is larger across the top-4 encoding layers. Interestingly, the dv^l is closer to zero for values of $f \geq 5$ across the second encoding layer compared to that of the models trained with $f \in [2, 3, 4]$. This implies that for these models with $f \geq 5$, the two *FI* units capture almost identical contextual information. Overall the models trained with $f \in [2, 3, 4]$ have the lowest divergence across the top three layers among all the values of *f* under consideration. In contrast, the models with $f \geq 5$ have the lowest divergence between the *FI* units across the first three layers.

Fig. 4b shows that, on average, the divergence across the entire encoding subnetwork is higher for smaller values of *f*. Specifically, the E_d for the models trained with $f \in [2, 3, 4]$ is greater than that of $f \geq 5$. This implies that each *FI* unit within the encoding layers for the models with $f \in [2, 3, 4]$ can capture more diverse contextual information when generating h_g and h_l. However, there is less diversity with $f \geq 5$. Overall, the variation of the divergence as shown in Fig. 4 and the corresponding translation quality displayed in Fig. 3 confirm our hypothesis that smaller values of *f* allow the *DC* module to effectively capture more diverse contextual information to further improve the performance of the translation model.

7.3 Layers to consider

Recent work (Belinkov et al. 2017; Peters et al. 2018; Raganato and Tiedemann 2018) has revealed that each encoder layer captures different levels of abstraction of the source sequence. Therefore, these layers tend to have different requirements for the local contextual information. Based on these findings, Yang et al. (2018), Xu

Fig. 5 BLEU scores on the En→De task for the Transformer baseline, the *DC* module-based models with respect to varying source sentence lengths



et al. (2019) and Shen et al. (2018) argue in favour of limiting the explicit localness modeling to the first few encoding layers, in order to allow the top-level layers to focus more on capturing the global contextual information. In contrast, this work argues that higher performance can be achieved when the local and global information modeling is applied across all encoder layers. To investigate this further, an ablation study was performed where the *DC* module is applied to different combinations of the encoding layers.

The translation performance achieved for the layer combinations is summarized in Table 5. As shown, all combinations of the encoder layers consistently outperform the Transformer baseline, further confirming the importance of learning both global and local contextual source information. For example, training the model with the *DC* module across only the lower 2 encoder layers (Row 3) produced a performance gain of about +0.61 BLEU over the baseline. Among the different combinations of layers, limiting the *DC* module to only lower-level layers (Rows 2-6) achieved higher translation quality compared to when employed across only the top-level layers (Rows 6, 7 and 8). The worst performance (28.47 BLEU) is obtained when the *DC* module is incorporated into only the top-2 encoding layers [5-6] (Row 7). Besides, the different combinations across the lower-level layers (Rows 2-6) produced fairly identical results with [1-1] (Row 2) and [1-4] (Row 5) achieving the best performance. This is consistent with the observations made by Shen et al. (2018), Yang et al. (2018) and Xu et al. (2019). However, our NMT model achieves the best translation performance when the *DC* module is applied to all the encoding layers.

7.4 Length

Previous work (Luong et al. 2015; Dou et al. 2018) argued that one of the major weaknesses of NMT models is the translation of long sentences. To achieve higher performance on source sentences of any arbitrary length, the encoder-decoder model should be able to effectively capture both the long-distance and short-range dependencies between the tokens (Dou et al. 2018). Following Luong et al. (2015), sentences of identical or equal length are grouped and the translation quality of the outputs from the models for each group is calculated. The comparison presented here is based on the following sentence length groups: <10, 10-20, 20-30, 30-40, 40-50, and >50. For each length group, the translation quality is evaluated for outputs from the models under consideration. Fig. 5 summarizes the impact of the length of the source sentence.

As shown, both the Transformer baseline and our *DC*-based models (Enc-DC, Dec-DC and Full-DC) display identical variation in the translation quality across the different source sentence lengths. This is true especially for sentences with length greater than or equal to 20 subword tokens. However, our *DC*-based models consistently outperform the baseline across the different sentence groups with greater than 10 subword tokens. The performance gain achieved by our models is attributed to the addition of the *DC* module which allows both the encoder (in the case of Enc-DC), the decoder (in the case of Dec-DC) or both (in the case of Full-DC) to effectively exploit both the local and global contextual information required to improve the generation of the target translation. For shorter sentences with fewer than ten subword tokens, the global contextual model of the self-attention module is shown to be effective enough for the target generation. In contrast, for longer source sentences, further improvement in translation quality is achieved when both global and local contextual modelling are employed as shown across the sentence groups 10-20, 20-30, 30-40, 40-50 and>50. Overall, the best performance across the sentence groups (greater than 10 tokens) is achieved when the *DC* module is applied to only the encoding subnetwork (i.e. Enc-DC), which supports our recommendation to limit the global and local modelling to only the encoder subnetwork.

8 Conclusion

This work proposes the *Dual Contextual* (DC) module to leverage local and global contextual information to improve translation performance of the Transformer model. Three possible applications of the *DC* module, namely Enc-DC, Dec-DC and Full-DC, were presented. The experimental results on the nine translation tasks demonstrate the effectiveness of the proposed *DC* module. Furthermore, the analyses performed suggests that:

- exploiting both the global and local contextual information is beneficial to the overall performance of the translation model.
- the best performance is achieved when the *DC* module is applied to only the encoding layers (i.e. Enc-DC). The decoding subnetwork employing the *DC* module (in the case of Dec-DC and Full-DC) produces lower performance gains over the baseline.
- in contrast to the findings of recent work (Shen et al. 2018; Yang et al. 2018; Xu et al. 2019), applying the *DC* module across all encoding layers results in higher performance gains compared to limiting its application to only the first few lower-level encoding layers (e.g. from layer 1 to 3).

Future work includes validating the performance of the proposed *DC* module on other NLP tasks such as document summarization, sequence tagging, and machine reading comprehension. Another interesting direction will consider investigating further the performance–model complexity trade-off of the application of the *DC* module. To further improve performance, we plan to extend the *DC* module to exploit the self-attention techniques presented in Song et al. (2018b), Yang et al. (2018) and Xu et al. (2019).

References

Al-Sabahi K, Zuping Z, Nadher M (2018) A hierarchical structured self-attentive model for extractive document summarization (HSSAS). IEEE Access 6:24205–24212

Ba JL, Kiros JR, Hinton GE (2019) Layer normalization. arXiv: 1607.06450, 2016

Bahdanau D, Cho K, Bengio Y (2015). Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations. San Diego, CA, pp 15

Belinkov Y, Màrquez L, Sajjad H, Durrani N, Dalvi F, Glass J (2017). Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In: Proceedings of the 8th international joint conference on natural language processing, Vol 1: Long Papers. Taipei, Taiwan, pp 1–10

Cettolo M, Niehues J, Stüker S, Bentivogli L, Federico M (2014) Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In: Proceedings of the international conference on spoken language translation, Lake Tahoe, CA, p 57

Cettolo M, Niehues J, Stüker S, Bentivogli L, Federico M (2015) The IWSLT 2015 evaluation campaign. In: Proceedings of the international conference on spoken language translation. Da Nang, p 13

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv: 1406.1078

Conneau A, Kruszewski G, Lample G, Barrault L, Baroni M (2018) What you can cram into a single vector: probing sentence embeddings for linguistic properties. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1: Long Papers. Melbourne, pp 2126–2136

Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: Proceedings of the 34th international conference on machine learning. Stockholm, pp 933–941

Dou Z-Y, Tu Z, Wang X, Shi S, Zhang T (2018) Exploiting deep representations for neural machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Brussels, pp 4253–4262

Dou Z-Y, Tu Z, Wang X, Wang L, Shi S, Zhang T (2019) Dynamic layer aggregation for neural machine translation with routing-by-agreement. arXiv:1902.05770,

Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN (2017) Convolutional sequence to sequence learning. In: Proceedings of the 34th international conference on machine learning. Sydney, pp 1243–1252

Gimpel K, Smith NA (2008) Rich source-side context for statistical machine translation. In: Proceedings of the third workshop on statistical machine translation, Columbus, Ohio, pp 9–17

He T, Tan X, Xia Y, He D, Qin T, Chen Z, Liu T-Y (2018) Layer-wise coordination between encoder and decoder for neural machine translation. Advances in Neural Information Processing Systems. Montreal, Canada, pp 7944–7954

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA, pp 770–778

He Z,Liu Q, Lin S (2008) Improving statistical machine translation using lexicalized rule selection. In: Proceedings of the 22nd international conference on computational linguistics (Coling 2008), Manchester, pp 321–328

Huang P-S, Wang C , Huang S, Zhou D, Deng L (2018) Towards neural phrase-based machine translation. In: Proceedings of the international conference on learning representations, p 14

Inan H, Khosravi K, Socher R (2016) Tying word vectors and word classifiers: aloss framework for language modeling. arXiv:1611.01462,

Kingma DP, Ba J (2014). Adam: amethod for stochastic optimization. arXiv: 1412.6980

Kitaev N, Klein D (2018) Constituency parsing with a self-attentive encoder. In: Proceedings of the 56th annual meeting of the association for computational linguistics. Melbourne, pp 2676–2686

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 conference on mpirical methods in natural language processing. Barcelona, pp 388–395

Li J, Yang B, Dou Z-Y, Wang X, Lyu MR, Z. Tu (2019). Information aggregation for multi-head attention with routing-by-agreement. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies. Minneapolis, MN, pp 3566–3575

Lin Z, Feng M, Santos CND, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. In:Proceedings of the international conference on learning representations, p 15

Luong M-TM , Manning CD (2015) Stanford neural machine translation systems for spoken language domains. In: Proceedings of the international conference on spoken language translation. Da Nang, pp 76–79

Luong M-T, Pham H, Manning CD (2015). Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Lisbon, pp 1412–1421

Marton Y, Resnik P (2008) Soft syntactic constraints for hierarchical phrased-based translation. In: Proceedings of association for computational linguistics: human language technologies. Columbus, Ohio, pp 1003–1011

Neubig G, Dou Z-Y, Hu J, Michel P, Pruthi D, Wang X (2019) Compare-mt: a tool for holistic comparison of language generation systems. In: Proceedings of the 2019 conference of the North American Chapter of the association for computational linguistics: human language technologies. Minneapolis, MN, pp 35–41

Papineni K, Roukos S, Ward T, Zhu W-J (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. Philadelphia, PA, pp 311–318

Pappas N, Miculicich L, Henderson J (2018). Beyond weight tying: Learning joint input-output embeddings for neural machine translation. In: Proceedings of the third conference on machine translation: research papers. Brussels, pp 73–83

Parikh A, Täckström O, Das D, Uszkoreit J (2016) A decomposable attention model for natural language inference. In: Proceedings of the 2016 conference on empirical methods in natural language processing. Austin, TX, pp 2249–2255

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American Chapter of the association for computational linguistics: human language technologies. New Orleans, LA, pp 2227–2237

Popel M, Bojar O (2018) Training tips for the transformer model. Prague Bull Math Linguist 110(43–70):2018

Raganato A, Tiedemann J (2018) An analysis of encoder representations in transformer-based machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing workshop BlackboxNLP: analyzing and interpreting neural networks for NLP. Brussels, pp 287–297

Sennrich R, Haddow B, Birch A (2016) Neural machine translation of rare words with subword units. In: Proceedings of the 54th annual meeting of the association for computational linguistics, vol 1: Long Papers. Berlin, pp 1715–1725

Shaw P, Uszkoreit J, Vaswani A (2018) Self-attention with relative position representations. In: Proceedings of the 2018 conference of the North American Chapter of the association for computational linguistics: human language technologies, vol 2 (Short Papers). New Orleans, LA, pp 464–468

Shen T, Zhou T, Long G, Jiang J, Zhang C (2018) Bi-directional block self-attention for fast and memory-efficient sequence modeling. In: Proceedings of the international conference on learning representations. Vancouver, p 18

So D, Le Q, Liang C (2019) The evolved transformer. International conference on machine learning. Long Beach, CA, pp 5877–5886

Song K, Tan X, He D, Lu J, Qin T, Liu T-Y (2018a) Double path networks for sequence to sequence learning. In: Proceedings of the 27th international conference on computational linguistics. Santa Fe, New Mexico, pp 3064–3074

Song K, Tan X, Peng F,Lu J(2018b) Hybrid self-attention network for machine translation. arXiv: 1811.00253,

Sperber M, Niehues J, . Neubig G , Stüker S, Waibel A (2018) Self-attentional acoustic models. In: Proceedings of interspeech 2018. Hyderabad, pp 3723–3727

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin Ł (2017) Attention is all you need. Advances in neural information processing systems. Long Beach, California, pp 5998–6008

Wang H, Ren J (2018) A self-attentive hierarchical model for jointly improving text summarization and sentiment classification. Asian conference on machine learning. Beijing, China, pp 630–645

Wu F, Fan A, Baevski A, Dauphin Y, Auli M (2019). Pay less attention with lightweight and dynamic convolutions. In: Proceedings of the international conference on learning representations. Vancouver, Canada, p 14

Wu Y, Schuster M, Chen Z, Le Q V, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv: 1609.08144

Xia Y, He T, Tan X, Tian F, He D, Qin T (2019) Tied transformers: Neural machine translation with shared encoder and decoder. In: Proceedings of the AAAI conference on artificial intelligence, vol 33. Honolulu, Hawaii, pp 5466–5473

Xu M, Wong DF, Yang B, Zhang Y, Chao L S (2019) Leveraging local and global patterns for self-attention networks. In: Proceedings of the 57th annual meeting of the association for computational linguistics, Florence, pp 3069–3075

Yang B, Tu Z, Wong D F, Meng F, Chao L S, Zhang T (2018). Modeling localness for self-attention networks. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Brussels, Belgium, pp 4449–4458

Yang B, Li J, Wong DF, Chao L S, Wang X, Tu Z (2019a). Context-aware self-attention networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 33. Honolulu, Hawaii, pp 387–394

Yang B, Wang L, Wong D F, Chao LS, Tu Z (2019b). Convolutional self-attention networks. In: Proceedings of the conference of the North American Chapter of the association for computational linguistics: human language technologies. Minneapolis, MN, pp 4040–4045

Yu A W, Dohan D, Luong M-T, . Zhao R, Chen K, Norouzi M, Le Q V (2018) QANet: Combining local convolution with global self-attention for reading comprehension. In: Proceedings of the international conference on learning representations. Vancouver, p 16

Zhang B, Xiong D, Su J (2018). Accelerating neural transformer via an average attention network. In: Proceedings of the 56th annual meeting of the association for computational linguistics. Melbourne, pp 1789–1798

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.