COOL, a Context Outlooker, and Its Application to Question Answering and Other Natural Language Processing Tasks

Fangyi Zhu, **See-Kiong Ng** and **Stéphane Bressan**
National University of Singapore
zhufy1030@gmail.com,{seekiong, steph}@nus.edu.sg

Abstract

Vision outlooker improves the performance of vision transformers, which implements a self-attention mechanism by adding an outlook attention, a form of local attention.

In natural language processing, as has been the case in computer vision and other domains, transformer-based models constitute the state-of-the-art for most processing tasks. In this domain, too, many authors have argued and demonstrated the importance of local context.

We present an outlook attention mechanism, COOL, for natural language processing. COOL, added on top of the self-attention layers of a transformer-based model, encodes local syntactic context considering word proximity and more pair-wise constraints than dynamic convolution used by existing approaches.

A comparative empirical performance evaluation of an implementation of COOL with different transformer-based models confirms the opportunity for improvement over a baseline using the original models alone for various natural language processing tasks, including question answering. The proposed approach achieves competitive performance with existing state-of-the-art methods on some tasks.

1 Introduction

Transformer neural networks, based solely on the self-attention mechanism [Vaswani *et al.*, 2017], have been shown to be remarkably successful in handling various tasks in natural language processing (NLP) [Lan *et al.*, 2019; He *et al.*, 2021] and other domains [Dosovitskiy *et al.*, 2021; Sun *et al.*, 2019; Dong *et al.*, 2018]. Self-attention, in a nutshell, considers every token in the input sequence when encoding a current token. This strategy allows the learning and capture of cross-passage long-range dependencies.

The interrogation naturally arises of whether self-attention is detrimental to local information and of how, if the case, this flaw can be remedied. Several authors have considered and evaluated various ways of incorporating local context information back into a self-attention mechanism for various tasks in various domains [Wu *et al.*, 2019; Gulati *et al.*, 2020; Yuan *et al.*, 2021; Xiao *et al.*, 2021].

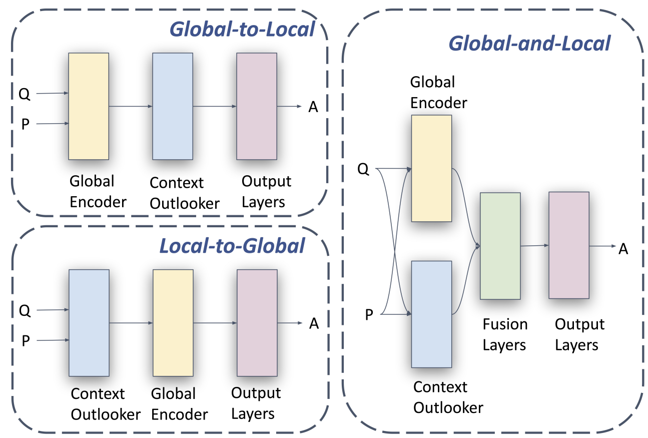


Figure 1: Different modes of integration of global and local encoding.

Recently, Yuan *et al.* [2021] proposed Vision Outlooker (VOLO) for visual recognition. VOLO successfully improves the performance of vision transformers by adding a light-weight outlook attention mechanism, a form of local attention.

We present and evaluate the outlook attention mechanism for NLP, named **Co**ntext **Outl**ooker (COOL). COOL implements a context outlook attention mechanism that encodes the local syntactic context considering word proximity and can be inserted into most transformer-based models. Compared to the operations for local augmentation used by existing works, outlook attention considers more pair-wise constraints as it processes multiple local windows containing an anchor token. We consider three possible modes for the composition of global and local encoding, in which the global encoding precedes, *Global-to-Local*, follows, *Local-to-Global*, or is parallel to, *Global-and-Local*, the local encoding.

Comparative empirical performance evaluation of COOL with different transformer-based approaches confirms the opportunity for improvement over a baseline using the original models for eleven NLP tasks, including question answering, question generation, sentiment analysis, etc.

2 Related Work

2.1 Transformer-based Language Models

Vaswani *et al.* [2017] first proposed the Transformer architecture, based solely on an attention mechanism, for neural machine translation (NMT). Since Bidirectional Encoder Representation from Transformer (BERT) [Devlin *et al.*, 2019] outstandingly tackled eleven NLP tasks, fine-tuning a transformer-based language model has become the mainstream approach for most NLP tasks. The transformer-based structure has been explored for other domains as well and achieved competitive performance with classical convolutional neural network- (CNN-) or recurrent neural network- (RNN-) based models, such as Vision Transformer (ViT) for computer vision (CV) [Dosovitskiy *et al.*, 2021], Bidirectional Encoder Representations from Transformers for Sequential Recommendation (BERT4Rec) for recommendation system [Sun *et al.*, 2019], and Speech-Transformer for automatic speech recognition (ASR) [Dong *et al.*, 2018].

The self-attention mechanism, an operation for cross-passage encoding derived from [Cheng *et al.*, 2016], is the crucial module in a transformer-based architecture. It relates all the elements in a sequence to capture long-range dependencies [Bahdanau *et al.*, 2014]. Nonetheless, there are two drawbacks of long-range encoding: the computation is expensive, especially for long sequences, and the cross-passage encoding might dilute the local information within the surrounding context. To address these issues, recent works from different domains proposed to exploit an adaptive span [Hofstätter *et al.*, 2020; Yang *et al.*, 2018; Sukhbaatar *et al.*, 2019], or augment local encoding, e.g., [Wu *et al.*, 2019; Zhang *et al.*, 2020; Jiang *et al.*, 2020] for NLP, [Liu *et al.*, 2021b; Yuan *et al.*, 2021] for CV, and [Gulati *et al.*, 2020] for ASR.

2.2 Local Attention

The dilution of local information in a standard self-attention mechanism for NLP and the benefit of its reinstatement was first discussed for NMT by Wu *et al.* [2019]. They presented the dynamic convolution to replace global encoding with the self-attention layers. They demonstrated that only encoding the local context could outperform executing a global encoding via self-attention.

Several works followed that tried to integrate both local, leveraging the dynamic convolution technique, and global encoding, leveraging the self-attention mechanism. Zhang *et al.* [2020] presented a novel module where a dynamic convolution and a self-attention mechanism were placed in parallel for NMT. Jiang *et al.* [2020] further connected a self-attention mechanism and a dynamic convolution in series to form a mixed-attention mechanism.

Recently, in the area of CV, Yuan *et al.* [2021] and Xiao *et al.* [2021] demonstrated that a significant factor limiting the performance of the transformer-based fashion for visual tasks was the low efficacy of the self-attention mechanism in encoding the local features as well. Moreover, Yuan *et al.* [2021] proposed VOLO for efficient pixel token aggregation via an outlook attention mechanism, with which a transformer-based model could outperform the classical CNN-based models on visual recognition and segmentation. Compared to the standard dynamic convolution technique, handling a single constraint within the current window only, the outlook attention mechanism deals with more pairwise constraints for the current token encoding. Specifically, it considers all local windows containing the current token.

3 Methodology

We present COOL, an outlook attention mechanism and architecture, that helps emphasise local information in NLP using a transformer-based model.

For the sake of clarity, we present the proposed model for a specific task of extractive question answering.

3.1 Architecture

Yuan *et al.* [2021] argued the importance of local information for visual tasks and proposed VOLO to overcome the weakening of local information in representations produced by the standard self-attention mechanism. The outlook attention mechanism, the primary component of VOLO, dynamically fine-tunes pixel token features produced by the self-attention mechanism considering local information from surrounding pixel tokens.

We present to generalise this idea into NLP as the context outlooker (COOL). The overall architecture has three major modules, global encoding by a generic transformer-based encoder, local encoding via the proposed context outlooker, and output layers for the downstream tasks. The output layers for an extractive question answering task are for answer prediction.

There are three different modes for integrating global and local encoding illustrated in Figure 1,

- *Global-to-Local (G2L)*: first encodes the question and passage by the global encoder. Then, the yielded global embeddings go through the context outlooker to enrich the local context information. Finally, the start and end positions of the answer are predicted in the output layers.
- *Local-to-Global (L2G)*: switches the order of global encoding and local augmentation in the first structure. In contrast to *G2L*, *L2G* first conducts local encoding, followed by global encoding.
- *Global-and-Local (G&L)*: executes global and local encoding parallelly and fuses the integrated global and local embeddings with several linear layers. Finally, predict the answer locations.

Next, we elaborate the workflow in light of the first structure of *G2L* in Figure 2.

3.2 Global Encoder

Given a question $q = \{q^1, ..., q^M\}$ and a passage $p = \{p^1, ..., p^N\}$, M, N denotes the total number of tokens in the question and passage. The global encoder, as a backbone of our proposed system, can be the encoder in any transformer-based model, e.g., BERT [Devlin *et al.*, 2019], ALBERT [Lan *et al.*, 2019], and RoBERTa [Zhuang *et al.*, 2021] with an

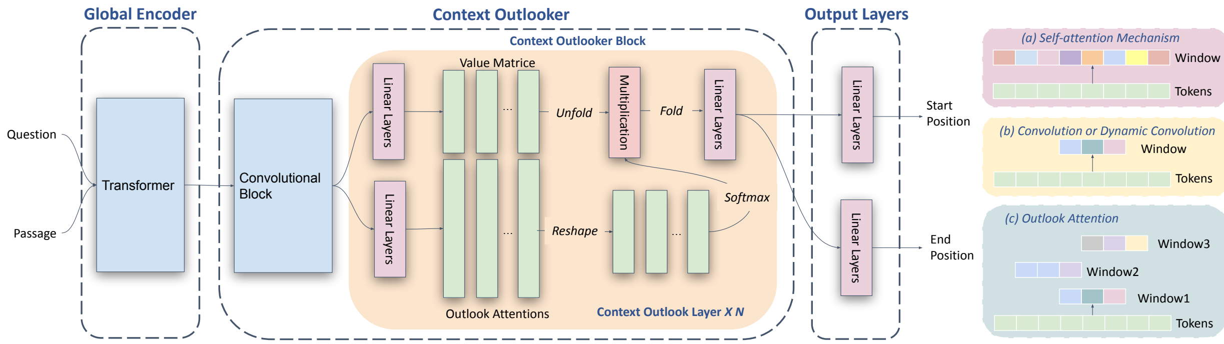


Figure 2: Diagram of our proposed system with a *G2L* structure. There are three major components: first, global encoding by a transformer-based encoder; second, local augmentation by the context outlooker; third, answer prediction by the output layers. *Unfold* and *Fold* correspond to unfold and fold operations in PyTorch [Paszke *et al.*, 2019]. [1]

encoder-only architecture, or ProphetNet [Qi *et al.*, 2020] with an encoder-decoder architecture.

$$h_g = Global([q; p])), \tag{1}$$

where $h_g \in \mathbb{R}^{L\times H}$ is the obtained global representations, L is the max length of the input sequence, and H is the size of hidden units in the last layer of the global encoder.

3.3 Context Outlooker

Context outlooker, the pivotal component in our proposed approach, consists of a convolutional block and a context outlook block.

Convolutional Block

The convolution operation is effective and stable for capturing local information [Kalchbrenner *et al.*, 2014]. Several works have proved that inserting convolutional layers can help transformers in applications to different domains [Wu *et al.*, 2019; Xiao *et al.*, 2021; Karpov *et al.*, 2020; Liu *et al.*, 2021a; Gulati *et al.*, 2020]. We thus try to insert a convolutional block for local augment, and further compare the convolution technique to our proposed module.

We stack the convolutional block, consisting of several convolutional layers, on top of the global encoder,

$$h_c = ReLU(ConvBlock(h_g)), \tag{2}$$

$$h_c = \left[h_c^1, ..., h_c^d, ...h_c^D\right], \tag{3}$$

$$\hat{h_c} = Concat(AdaptivePool(h_c)). \tag{4}$$

In Eq. (3), h_c^d is the output of the d-th convolutional layer with ReLU non-linear activation operation [Glorot *et al.*, 2011], and D is the number of convolutional layers in the block. The dimension of h_c^d is $L^d \times F^d$. L^d depends on the kernel size and padding size of the convolutional layer, and F^d is the number of filters in the convolutional layer. To align different L^d to original length L, we perform adaptive pooling after each convolutional layer interpreted in Eq.(4). Finally, the outputs from all convolutional layers are concatenated along the second dimension to produce $\hat{h_c} \in \mathbb{R}^{L\times F}$,

$$F = \sum_{1\le d<D} F^d. \tag{5}$$

[1]Unfold extracts sliding local blocks from a batched input tensor. Fold, the reverse operation of unfold, combines an array of sliding local blocks into the original large containing tensor.

Context Outlook Block

The context outlook block, consisting of several context outlook layers, follows by the convolutional block.

In analogy with the feature map of an image, each row in the representation matrix $\hat{h_c}$ is the representation of a token. To maintain the integrality of representation of each token, we adapt the local window size in the visual outlook attention mechanism to $K \times F$ to handle text tokens instead of $K \times K$ for processing image pixel tokens.

Similar to the computation of *Query*, *Key*, *Value* matrices in the self-attention mechanism, the outlook attention mechanism first produces value matrix $v \in \mathbb{R}^{L\times F}$ via a linear layer with $\hat{h_c}$ from previous convolutional block,

$$v = W_v\hat{h_c} + b_v, \tag{6}$$

where W_v and b_v separately denote the parameters and bias needed to be learned in the linear layer.

Hence, for the i-th token, the value matrices of its neighbour tokens in a local window with the size of $K \times F$ are,

$$v_{\Delta_i} = v_{i+r-\lfloor \frac{K}{2} \rfloor}, \quad 0 \le r < K, \tag{7}$$

$v_{\Delta_i} \in \mathbb{R}^{K\times F}$, and $\lfloor\cdot\rfloor$ is the round down formula.

In visual outlook attention, Yuan *et al.* [2021] proposed to produce the aggregating weights directly from the features of the centric token to dispense the expensive dot product computation in self-attention, as each spatial location in an image feature map can represent its close neighbours. Likewise, the embedding of a token in a sentence is able to express its neighbour tokens as well. Therefore, following their work, we calculate attention matrix $a \in \mathbb{R}^{L\times A}$ with $\hat{h_c}$ via another linear layer,

$$a = W_a\hat{h_c} + b_a, \tag{8}$$

here $A = K \times K \times F$. W_a and b_a are learned parameters and bias. Then, the attention weights within different local windows are obtained by reshaping a from $\mathbb{R}^{L\times A}$ to $\mathbb{R}^{L\times K\times(K\times F)}$, $a_{\Delta_i} \in \mathbb{R}^{1\times1\times(K\times F)}$ represents the obtained attention weights of i-th token within a local window.

Therewith, normalise the attention weights in each local window with a softmax function and aggregate the neighbours within the window with the acquired attention weights,

$$\hat{v}_{\Delta_i} = softmax(a_{\Delta_i})v_{\Delta_i}. \tag{9}$$

Afterward, sum up the weighted values at the same position from different windows,

$$\tilde{v}_i = \sum_{0 \le r < K} \hat{v}^{i}_{\Delta_{i+r-\lfloor \frac{K}{2} \rfloor}}, \tag{10}$$

where $\tilde{v}_i$ is the acquired representation for i-th token with rich local context information, and $\tilde{v}$ is the representation matrice for the whole sequence.

Finally, conduct a residual operation [He *et al.*, 2016] that sum the encoded $\tilde{v}$ with $\hat{h_c}$, and go through several linear layers for further representation learning, followed by performing another residual operation,

$$h_f = \tilde{v} + \hat{h_c}, \tag{11}$$

$$\hat{h}_f = W_f h_f + b_f, \tag{12}$$

$$h_f^{'} = \hat{h}_f + h_f. \tag{13}$$

3.4 Output Layers

We stack linear layers to perform answer prediction with the obtained $h_f^{'}$,

$$y_s = \mathrm{softmax}(W_s h_f^{'} + b_s), \tag{14}$$

$$y_e = \mathrm{softmax}(W_e h_f^{'} + b_e), \tag{15}$$

W_s, W_e and b_s, b_e are parameters and bias needed to be learned in the linear layers. y_s, y_e are the probability distributions for the start position and the end position.

The training objective is to minimize the sum of the negative log probabilities at the ground truth start position a_s and end position a_e,

$$\mathcal{L} = -\log(y_s^{a_s} + y_e^{a_e}). \tag{16}$$

4 Experiments

We evaluate the proposed system for extractive question answering on the Stanford Question Answering Dataset (SQuAD), i.e., SQuAD 2.0 [Rajpurkar *et al.*, 2018] and SQuAD 1.1 [Rajpurkar *et al.*, 2016], with no additional training data. We use the standard metrics for question answering, F1 measure and exact match (EM).

We conduct extensive experiments with other NLP tasks and other language to evaluate our proposed approach's versatility and significance.

4.1 Setup

We leverage the encoder in BERT-base[2], ALBERT-base[3], RoBERTa-base[4], and ProphetNet[5] for text generation as the global encoder and keep their default settings. The max sequence length in our experiments is set to 384.

In the convolutional block, we stack multiple convolutional layers to augment local information within different n-grams. We stack $3 \times (H + 4)$, $4 \times (H + 4)$, and $5 \times (H + 4)$ convolutional layers with 100 filters each. H is the dimension of the input representations, for instance, H is 768 while using BERT-base as global encoder. 2-padding is used to maintain the sequence length.

[2]*bert-base-cased*: https://huggingface.co/bert-base-cased
[3]*albert-base-v2*: https://huggingface.co/albert-base-v2
[4]*roberta-base-squad2*: https://huggingface.co/roberta-base
[5]https://github.com/microsoft/ProphetNet

Architecture	Outlook Layers	F1	EM
BERT	-	75.41	71.78
G2L w/o Conv.	3	76.79	73.36
G2L w/ Conv.	2	**77.11**	**74.02**
L2G w/o Conv.	3	74.80	72.02
L2G w/ Conv.	2	50.06	49.98
G&L w/o Conv.	3	75.58	72.78
G&L w/ Conv.	2	75.98	72.86

Table 1: SQuAD 2.0 Dev results of different integration modes.

In the context outlook block, the kernel size of context outlook layers is 3×300, depending on the number of filters in the convolutional layers. While without a convolutional block, it is set to $3 \times H$ according to the global encoder.

During training, we use AdamW [Loshchilov and Hutter, 2019] optimizer. We set a learning rate to 2e-5 for the global encoder, and 1e-4 for other layers. We train the model with a mini-batch size of 48 for about 5 epochs. The main codes are uploaded as supplementary materials.

4.2 Extractive Question Answering

Choice of architecture. We first compare the three modes of global and local encoding integration: *G2L*, *L2G*, and *G&L*.

Table 1 reports the results of the respective architectures with BERT as a baseline. The table shows the most effective mode is *G2L*. *G&L* achieves a minor improvement over the baseline, yet not exceeding that of the first structure. A possible reason is it's difficult to precisely capture the relationships between global and local representations. We find *L2G* architecture does not work well. The local encoding operation perturbs the following global encoding procedure, particularly while integrating the convolution operations with different kernel sizes. Specifically, directly combining the information within different n-grams confuses to the model. In summary, it should better not replace the input to the pretrained model. We nevertheless acknowledge that early convolution might benefit a scenario as discussed in [Xiao *et al.*, 2021] for visual tasks. In the following experiments, we adopt the *G2L* mode.

Overall results. Table 2 shows the comparisons of COOL with and without the convolutional block combined with BERT, ALBERT, RoBERTa and the corresponding baselines for SQuAD 2.0 and SQuAD 1.1. It can be viewed that COOL with BERT achieves an improvement of 2.24 and 1.7 points for EM and F1, respectively, over the original BERT. COOL with ALBERT adds 0.88 EM points and 0.86 F1 points. COOL with RoBERTa outperforms the original in terms of both EM and F1 as well. SQuAD 1.1 does not contain unanswerable questions and is thus less challenging than SQuAD 2.0. Nevertheless, models inserted COOL improve across the board over the original models. These results confirm the benefit of an appropriate consideration of local information

Model	F1	EM
SQuAD 2.0		
BERT	75.41	71.78
COOL(BERT) w/o Conv.	76.79	73.36
COOL(BERT) w/ Conv.	**77.11**	**74.02**
ALBERT	78.63	75.33
COOL(ALBERT) w/o Conv.	79.23	75.95
COOL(ALBERT) w/ Conv.	**79.49**	**76.21**
RoBERTa	82.91	79.87
COOL(RoBERTa) w/o Conv.	83.39	80.12
COOL(RoBERTa) w/ Conv.	**83.61**	**80.33**
SQuAD 1.1		
BERT	88.33	80.76
COOL(BERT) w/o Conv.	88.38	**81.00**
COOL(BERT) w/ Conv.	**88.44**	80.91
ALBERT	87.81	79.50
COOL(ALBERT) w/o Conv.	87.81	79.93
COOL(ALBERT) w/ Conv.	**87.90**	**80.15**
RoBERTa	90.40	83.00
COOL(RoBERTa) w/o Conv.	90.65	83.36
COOL(RoBERTa) w/ Conv.	**90.66**	**83.42**

Table 2: SQuAD Dev results for extractive question answering.

Model	F1	EM
BERT	75.41	71.78
BERT-Deeper(3 layers)	75.28	71.81
BERT-Deeper(5 layers)	74.94	71.28
COOL(BERT) w/o Conv.	76.79	73.36
COOL(BERT) w/ Conv.	**77.11**	**74.02**

Table 3: Comparisons with deeper models.

Model	F1	EM
BERT	75.41	71.78
BERT + Convolution	75.82	72.81
BERT + Dynamic Convolution	75.57	71.19
BERT + Visual Outlook Attention	75.89	72.94
BERT + Context Outlook Attention	**76.79**	**73.36**

Table 4: Comparisons with other techniques for local augmentation.

in a transformer-based approach for question answering that COOL is an effective approach.

The comparison of the results for COOL with and without the convolutional block, further shows that COOL with the convolutional block generally outperforms COOL without the convolution block, except for the case of SQuAD 1.1 where COOL(BERT) without the convolutional block marginally outperforms COOL(BERT) with the convolutional block by 0.09 point. In the following ablation study, we further explore and compare the impacts of the convolution techniques and the outlook attention.

Ablation study. To verify these improvements are contributed by COOL rather than more trainable parameters, we first compare the performance with a deeper BERT containing more self-attention layers in Table 3. We added 3 or 5 more self-attention layers on top of the original BERT. It can be observed that direct stacking layers does not gain an improvement over the base model, even bringing adverse effects. It further confirms that COOL brings these improvements and it can remedy the deficiencies of the self-attention mechanism while encoding the local information.

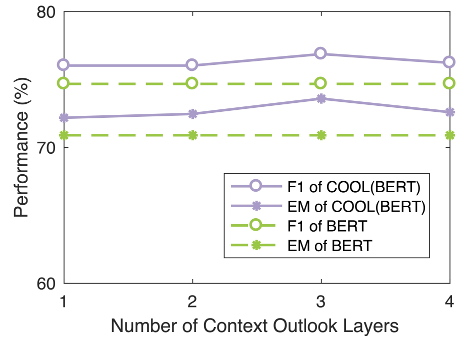


Figure 3: Results with different numbers of context outlook layers.

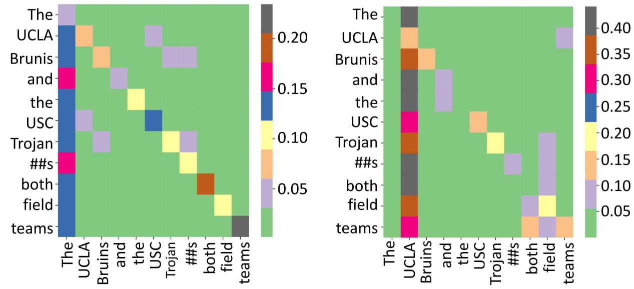
We further compare the outlook attention with other techniques for local information enhancement in Table 4. We separately implement BERT with standard convolution and dynamic convolution [Wu *et al.*, 2019; Jiang *et al.*, 2020]. The table reports the COOL(BERT) results without the convolutional block in the interest of fairness. We can see that the models with outlook attention outperform those with standard or dynamic convolution.

We also compare the results between the original visual outlook attention and context outlook attention to demonstrate that the changes to the visual outlook attention to applying to NLP are relevant and effective. The results are collated in Table 4 as well. It can be viewed that outlook attention, whether implemented as the visual outlook or improved into a mechanism dedicated to NLP, brings an improvement. Thus, the modified module is indeed better suited to NLP tasks.

The results of COOL with different numbers of context outlook layers are reported in Figure 3. All variants with COOL outperform the baseline. It achieves peak performance with two context outlook layers with a convolutional block.

Illustrative examples. We first present and discuss illustrative examples of the situations in which models with COOL predict the correct or better answers while the baseline models do not and reverse. The examples indicate the question (Q), the paragraph or the significant excerpt thereof (P), the answers by the baseline models, and the answers by COOL with the corresponding baseline models. The key portions of the paragraph pertinent to the discussion are highlighted in bold, and correct answers are indicated with underlining.

In Example 1, the baseline models fail to predict the correct answer. After inserting COOL, COOL(BERT) and COOL(RoBERTa) precisely recognise the relationship between ‘USC’ and ‘Trojans’ and thus give the correct answer. Although COOL(ALBERT) still fails, it attempts to recognise the relationship between the state ‘UCLA’ and the team’s name ‘Bruins’. We further visualize the self-attention weights in Figure 4. We can observe the most important tokens to ‘USC’ are itself, ‘UCLA’, ‘both’, in that order. For ‘UCLA’, the important tokens are itself, ‘USC’, ‘the’, ‘Bruins’, ‘and’, and ‘both’. Likewise, the important words to ‘Bruins’ are



(a) BERT (b) RoBERTa

Figure 4: Visualization of weights yielded by the self-attention mechanism.

itself, 'both', 'and', 'Trojan', and to 'Trojan' are itself, its subtoken '##s', and 'Bruins'. It explains that the standard self-attention operation via similarity score calculation can not distinguish these tokens and understand their relationships. It thus predicts a long span as the answer. Whereas augmenting the local context information by COOL helps to address this disadvantage and benefits understanding the relations between close neighbours.

Example 1.
Q: What is the ***name of the team from USC****?*
P: ***The UCLA Bruins and the USC Trojans*** *both field teams...*
BERT: UCLA Bruins and the USC Trojans
ALBERT: UCLA Bruins
RoBERTa: USC Trojans
COOL(BERT): Trojans
COOL(ALBERT): Bruins
COOL(RoBERTa): Trojans

Moreover, we showcase interesting Example 2, the consideration of the local context allows COOL to capture logical relations where the baseline models fail. It is reasonable as most logical formulas are within local content.

Example 2.
Q: What is an expression that can be used to illustrate the suspected inequality of complexity classes?
P: Many known complexity classes are suspected to be unequal,... For instance $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PP} \subseteq \mathbf{PSPACE}$*, but it is possible that P = PSPACE.*
BERT: ⟨no answer⟩
ALBERT: P = PSPACE
RoBERTa: P = PSPACE
COOL(BERT): $P \subseteq NP \subseteq PP \subseteq PSPACE$*, but it is possible that P = PSPACE.*
COOL(ALBERT): $P \subseteq NP \subseteq PP \subseteq PSPACE$*,*
COOL(RoBERTa): $P \subseteq NP \subseteq PP \subseteq PSPACE$

There are, unfortunately, situations in which too much local attention can be misleading and detrimental, as illustrated by Examples 3. We target to address this challenge of balancing the global and local encoding in future work.

Example 3.
Q: What was the Anglo-Norman language's final form?
P: The ***Anglo-Norman*** *language was eventually absorbed into the Anglo-Saxon language of their subjects (see Old English) and influenced it, helping (along with the Norse language of the earlier Anglo-Norse settlers and the Latin used by the church) in the development of Middle English. It in turn evolved into* ***Modern English****.*
BERT: Modern English
ALBERT: ⟨no answer⟩
RoBERTa: Modern English
COOL(BERT): ⟨no answer⟩
COOL(ALBERT): ⟨no answer⟩
COOL(RoBERTa): Modern English

Model	B@4	M	R
ProphetNet	23.91	26.60	52.26
ERNIE-GEN (beam=1)	24.03	26.31	52.36
ERNIE-GEN (beam=5)	25.40	26.92	52.84
COOL(ProphetNet) (beam=1)	24.33	26.24	52.53
COOL(ProphetNet) (beam=5)	**25.93**	**27.14**	**52.86**

Table 5: SQuAD 1.1 Dev results for question generation.

Model	MNLI	QNLI	MRPC	RTE	STS-B
BERT	84.3	90.6	88.0	67.8	88.9
COOL(BERT)	84.3	**91.3**	**91.9**	**71.9**	**89.6**
RoBERTa	87.6	92.8	90.2	78.7	**91.2**
COOL(RoBERTa)	**87.8**	**93.0**	**93.3**	**81.3**	90.6

Table 6: GLUE Test results. F1 scores are reported for MRPC. Spearman correlations are reported for STS-B. Accuracy scores are reported for the other tasks.

4.3 Other Natural Language Processing Tasks

We seek to confirm the versatility of COOL and its effectiveness in tackling more NLP tasks. We compare the baseline model and a COOL-augmented version, noted COOL (⟨*base*⟩), for question generation, multiple-choice question answering, natural language inference, sentiment analysis, named-entity recognition, and part-of-speech tagging with the corresponding widely used datasets.

Question generation. Question generation (QG) targets generating a question given a passage and an answer [Qi *et al.*, 2020]. We evaluate the model with SQuAD 1.1 and adopt the commonly used metrics, BLEU@4 (B@4) [Papineni *et al.*, 2002], METEOR (M) [Banerjee and Lavie, 2005], and ROUGE-L (R) [Lin, 2004]. ERNIE-GEN [Xiao *et al.*, 2020] and ProphetNet [Qi *et al.*, 2020] achieve the state-of-the-art for QG. We insert COOL into ProphetNet, then compare it with the original ProphetNet and with ERNIE-GEN. The results of the different methods are compared in Table 5. The original ProphetNet has a BLEU@4 score of 23.91 and a METEOR score of 26.60. With COOL, the BLEU@4 and METEOR reach 25.93 and 27.14, significantly outperforming both the original ProphetNET and ERNIE-GEN.

Natural language inference. Natural language inference (NLI), also termed natural language understanding (NLU), targets to determine whether two input sequences entail or contradict each other [Wang *et al.*, 2018]. The General Language Understanding Evaluation benchmark (GLUE), composed of 9 datasets for different tasks, is widely used for NLI. We evaluate the proposed approach on the Multi-Genre Natural Language Inference Corpus (MNLI) [Williams *et al.*, 2018], the Question-answering NLI dataset (QNLI) [Rajpurkar *et al.*, 2016], the Microsoft Research Paraphrase Corpus (MRPC) [Dolan and Brockett, 2005], the Recognizing Textual Entailment dataset (RTE) [Bentivogli *et al.*, 2009], and the Semantic Textual Similarity Benchmark (STS-B) [Cer *et al.*, 2017]. The results are presented in Table 6. COOL(BERT) and COOL(RoBERTa) always outperform the originals, excluding on the MNLI dataset. More-

Model	SWAG	NER	POS	SST	IMDB
BERT	79.7	90.8	93.4	**92.5**	93.4
COOL(BERT)	**79.9**	**91.4**	**93.7**	92.2	**93.6**
RoBERTa	83.3	91.9	92.8	94.8	**91.3**
COOL(RoBERTa)	**83.4**	**95.3**	**93.5**	**95.2**	91.2

Table 7: SWAG Test accuracy results, CoNLL-2003 NER and POS F1 results, and SST and IMDB accuracy results.

over, COOL(BERT) and COOL(RoBERTa) achieve significant improvements on the MRPC and RTE datasets.

Multiple-choice question answering. Multiple-choice question answering (MCQA) is a type of retrieval-based question answering that selects an answer to a question by ranking the given choices. The Situations With Adversarial Generations (SWAG) dataset contains 113k sentence-pair examples that evaluate grounded commonsense inference [Zellers *et al.*, 2018]. It targets to choose the most plausible continuation among four choices. The results of the comparisons of the baselines with models with COOL are reported in Table 7. According to the results, COOL module is able to improve this task.

Named-entity recognition and part-of-speech tagging. Named-entity recognition (NER) locates and classifies named entities in a sentence into pre-defined categories. It extracts, for instance, names of locations and people from the text and places them under certain categories, such as organization and person [Yamada *et al.*, 2020]. Part-of-speech (POS) tagging assigns to each word in the text its part of speech, e.g., noun, verb, adjective, and other grammatical categories such as tense [Tian and Lo, 2015]. NER and POS tagging are basic NLP tasks. We evaluate the model performance on the CoNLL-2003 dataset [Tjong Kim Sang and De Meulder, 2003]. The comparative results are reported in Table 7. We can observe that COOL benefits the two tasks as well.

Sentiment analysis. Sentiment analysis (SA) determines whether a sentence is positive, negative, or neutral. The common datasets are the large movie review dataset (IMDB) [Maas *et al.*, 2011] and the Stanford Sentiment Treebank (SST-2) from GLUE. The comparisons are also presented in Table 7. For this task, the results are mixed, suggesting the possible lesser importance of the fine-grained element brought by involving the local context.

4.4 Other Language

To verify the proposed context outlooker is generalisable for other languages. We try to apply it to Malay, an important low-resource language in Southeast Asia. We evaluate the approach on Malay SQuAD 2.0 which is translated from English SQuAD 2.0 by Husein[6]. Compared with the original BERT-Bahasa[7] in Table 8, the proposed approach also brings improvement to the Malay language. It further proves the effectiveness and generalisability of the proposed module.

[6]https://github.com/huseinzol05/Malay-Dataset

[7]https://github.com/huseinzol05/malaya

Model	EM	F1
BERT-Bahasa	57.17	61.49
COOL(BERT)	**58.41**	**63.24**

Table 8: Malay SQuAD 2.0 Dev results for Malay extractive question answering.

Model	F1	EM
BERT	75.41	71.78
+ Local Graph (RGCN)	74.83	71.52
+ Local Graph (GAT)	75.79	**72.90**
+ Local Graph (HGT)	**75.96**	72.67

Table 9: Comparisons with GNNs on SQuAD 2.0.

4.5 Comparison With Augmentation by a Local Graph

We consider whether we can perform local augmentation by building a local graph on top of the fully connected graph processed by the self-attention mechanism. Thus, we try to build a local graph with the outputs from a backbone where only close neighbours within the predefined range are connected and adopt graph neural networks (GNNs) to update the vertex embeddings. In Table 9, we report the preliminary results by utilising different GNNs, i.e., relational graph convolution network (RGCN) [Michael Sejr *et al.*, 2018], graph attention network (GAT) [Veličković *et al.*, 2017], and heterogeneous graph transformer (HGT) [Hu *et al.*, 2020]. The model with RGCN underperforms the baseline as it treats all neighbours equally, which might weaken the discrimination of different neighbours. The models with GAT or HGT, considering weighted effects from different neighbours, outperform the baselines that prove the impact of local augmentation. Nevertheless, the results do not exceed the model with a context outlooker. We consider the reason is the GNNs handle the direct neighbours only. However, the context outlooker considers more constraints from indirect neighbours.

5 Conclusion

We presented COOL, an outlook attention mechanism for natural language processing that can be added to a transformer-based network for any natural language processing task.

A comparative empirical performance evaluation of an implementation of COOL with different models confirms that COOL, in its *Global-to-Local* mode, yields a practical performance improvement over a baseline using the original model alone for question answering. The performance and examples suggest that COOL is indeed able to acquire a more expressive representation of local context information.

We also presented empirical results for other natural language processing tasks: question generation, multiple-choice question answering, natural language inference, sentiment analysis, named-entity recognition, and part-of-speech tagging not only illustrate its portability and versatility of COOL over different models and for different tasks but also confirms the positive contribution of COOL to the effectiveness of the resolution of these tasks and other languages.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[Bentivogli *et al.*, 2009] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*, 2009.

[Cer *et al.*, 2017] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[Cheng *et al.*, 2016] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *the Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[Dolan and Brockett, 2005] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *the Workshop on Paraphrasing*, 2005.

[Dong *et al.*, 2018] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *the Conference on Acoustics, Speech and Signal Processing*, 2018.

[Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *The Ninth International Conference on Learning Representations*, 2021.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *the Conference on Artificial Intelligence and Statistics*, 2011.

[Gulati *et al.*, 2020] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Proc. Interspeech*, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[He *et al.*, 2021] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. *ArXiv*, abs/2006.03654, 2021.

[Hofstätter *et al.*, 2020] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. Local self-attention over long text for efficient document retrieval. In *the 43rd International Conference on Research and Development in Information Retrieval*, 2020.

[Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference*, 2020.

[Jiang *et al.*, 2020] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. ConvBERT: Improving BERT with span-based dynamic convolution. In *Advances in Neural Information Processing Systems*, 2020.

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *the Association for Computational Linguistics*, 2014.

[Karpov *et al.*, 2020] Pavel Karpov, Guillaume Godin, and Igor V Tetko. Transformer-CNN: Swiss knife for qsar modeling and interpretation. *Journal of Cheminformatics*, 2020.

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[Liu *et al.*, 2021a] Yun Liu, Guolei Sun, Yu Qiu, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. Transformer in convolutional neural networks. *arXiv preprint arXiv:2106.03180*, 2021.

[Liu *et al.*, 2021b] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *the Conference on Learning Representations*, 2019.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *the Association for Computational Linguistics*, 2011.

[Michael Sejr *et al.*, 2018] Schlichtkrull Michael Sejr, Kipf Thomas N., Bloem Peter, Berg Rianne Van Den, Titov Ivan, and Welling Max. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference*, 2018.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *the Association for Computational Linguistics*, 2002.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 2019.

[Qi *et al.*, 2020] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. ProphetNet: Predicting future n-gram for sequence-to-Sequence Pre-training. In *Findings of the Association for Computational Linguistics*, 2020.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *the Conference on Empirical Methods in Natural Language Processing*, 2016.

[Rajpurkar *et al.*, 2018] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *the Association for Computational Linguistics*, 2018.

[Sukhbaatar *et al.*, 2019] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In *the Association for Computational Linguistics*, 2019.

[Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *the Conference on Information and Knowledge Management*, 2019.

[Tian and Lo, 2015] Yuan Tian and David Lo. A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. In *the Conference on Software Analysis, Evolution, and Reengineering*, 2015.

[Tjong Kim Sang and De Meulder, 2003] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *the Seventh Conference on Natural Language Learning at HLT-NAACL*, 2003.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wang *et al.*, 2018] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *the EMNLP Workshop BlackboxNLP*, 2018.

[Williams *et al.*, 2018] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *the Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[Wu *et al.*, 2019] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations*, 2019.

[Xiao *et al.*, 2020] Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *arXiv preprint arXiv:2001.11314*, 2020.

[Xiao *et al.*, 2021] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

[Yamada *et al.*, 2020] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *the Conference on Empirical Methods in Natural Language Processing*, 2020.

[Yang *et al.*, 2018] Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. Modeling localness for self-attention networks. In *the Conference on Empirical Methods in Natural Language Processing*, 2018.

[Yuan *et al.*, 2021] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. VOLO: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021.

[Zellers *et al.*, 2018] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *the Conference on Empirical Methods in Natural Language Processing*, 2018.

[Zhang *et al.*, 2020] Zhebin Zhang, Sai Wu, Gang Chen, and Dawei Jiang. Self-attention and dynamic convolution hybrid model for neural machine translation. In *IEEE International Conference on Knowledge Graph (ICKG)*, 2020.

[Zhuang *et al.*, 2021] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In *the 20th Chinese National Conference on Computational Linguistics*, 2021.