applied sciences

MDPI

Article

Captioning Transformer with Stacked Attention Modules

Xinxin Zhu [1,2,†,‡], Lixiang Li [1,2,‡,*], Jing Liu [3,‡], Haipeng Peng [1,2,‡] and Xinxin Niu [1,2,‡]

[1] Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; zhuxinxin@bupt.edu.cn (X.Z.); penghaipeng@bupt.edu.cn (H.P.); xxniu@bupt.edu.cn (X.N.)

[2] National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China

[3] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; jliu@nlpr.ia.ac.cn

* Correspondence: li_lixiang2006@163.com; Tel.: +86-010-6228-2264

† Current address: School of Cyberspace Security, Beijing University of Posts and Telecommunications, P.O. Box 145, Haidian District, Beijing 100876, China.

‡ These authors contributed equally to this work.

Received: 30 March 2018; Accepted: 3 May 2018; Published: 7 May 2018

Abstract: Image captioning is a challenging task. Meanwhile, it is important for the machine to understand the meaning of an image better. In recent years, the image captioning usually use the long-short-term-memory (LSTM) as the decoder to generate the sentence, and these models show excellent performance. Although the LSTM can memorize dependencies, the LSTM structure has complicated and inherently sequential across time problems. To address these issues, recent works have shown benefits of the Transformer for machine translation. Inspired by their success, we develop a Captioning Transformer (CT) model with stacked attention modules. We attempt to introduce the Transformer to the image captioning task. The CT model contains only attention modules without the dependencies of the time. It not only can memorize dependencies between the sequence but also can be trained in parallel. Moreover, we propose the multi-level supervision to make the Transformer achieve better performance. Extensive experiments are carried out on the challenging MSCOCO dataset and the proposed Captioning Transformer achieves competitive performance compared with some state-of-the-art methods.

Keywords: image caption; image understanding; deep learning; computer vision

1. Introduction

The target of image captioning is to describe the content of images automatically. However, an image may contain various objects and these objects may have complex relations. This makes the image captioning become a difficult task. With the recent development of deep learning, more and more image captioning methods [1,2] have shown satisfactory results. Moreover, the descriptions generated by these models are closer to the natural language. The image captioning method based on the neural network usually contains the encoder and the decoder. To learn the meaning of an image, Convolutional Neural Network (CNN) is regarded as an encoder which can extract the semantic information in the image.RNN is the decoder which can decode the image feature into a text sequence. Nowadays, the long short-term memory (LSTM) and the gated recurrent neural (GRU) networks are the mainly used Recurrent Neural Networks (RNN) model.

However, RNN has the complex addressing and overwriting mechanism combined with inherently sequential processing problems. These pose challenges during training. For example, the LSTM has the hidden state h_t to memorize the historical information. To generate the current hidden state, it needs the previous hidden state h_{t-1} as the input. This mechanism is designed to make a good relationship across different time, but it also leads to the sequence training problem. The training time will increase with the increment of the sequence length, which inherently influences the training in parallel.

The attention mechanism has become an important method for the sequence-to-sequence problem. This mechanism can memorize the relation between the input and the output. Recently, the attention mechanism has been used in the RNN network and it has shown excellent performance. As is shown in the Soft-Attention model [2], the attention can help the model to attend to the salient objects in the image. The Transformer [3] model contains the stacked attention mechanism, eschewing recurrence. This mechanism can draw global dependencies between input and output. The Transformer has the self-attention and multi-head attention modules. The self-attention can correlate different positions and compute a representation for the whole sequence, while the multi-head attention can correlate different multi-modal representations and establish contact with the image and the text.

The Transformer model does not rely on the previous time result in training time. For example, to generate the current word, the model only need the previous ground-truth word, and we do not need the previous state h_{t-1} which is used in LSTM. The previous state h_{t-1} make a problem for the training, because the following state depends on the previous state. The Transformer model only contains stack attentions model and Feed Forward model, This structure is similar to the Multi-Layer Perceptron without time dependence.

The image captioning task also can be regarded as a sequence problem. Different from the machine translation, the source language becomes an image. This problem can be viewed as translating an image to the target sentence. So the encoder becomes a CNN which can recognize the meaning of an image. Based on the Transformer architecture, we proposed the Caption Transformer model (CT). Different from the original model, we discard the RNN as the decoder, in contrast, we use the Transformer as our decoder model.

Our key contributions are presented as follows: (a) A Captioning Transformer model that shows comparable performance to an LSTM-based method on standard metrics; (b) For better training the Transformer, a multi-level supervision training method is proposed to improve the performance. (c) We evaluate our architecture on the challenging MSCOCO dataset, and compare it with the LSTM and the LSTM+Attention baseline.

2. Related Work

2.1. Image Captioning

Inspired by the success of deep neural networks in Neural Machine Translation, the encoder-decoder framework has been proposed for image caption [1,4]. Vinyals et al. [1] firstly proposed an encoder-decoder framework, which used the CNN as the image encoder and the RNN as the sentence decoder. Further, various improvement methods have been developed. Jia et al. [5] used the semantic information to guide the LSTM along the sentence generation. Xu et al. [2] proposed a spatial attention mechanism to attend to different parts of the image dynamically. Yang et al. [6] proposed a review network to extend the existing encoder-decoder models. [7,8] fed the attribute features into RNNs to leverage the high-level attributes. Anderson et al. [9] proposed a combined bottom-up and top-down attention mechanism based on the object detection methods.

2.2. Transformer

Nowadays, the sequence models are usually based on the RNN model or the CNN model, and they include an encoder and a decoder. An attention mechanism is used to connect the encoder and the decoder, and it has achieved better performance. Vaswani et al. [3] proposed a simple network architecture, i.e., the Transformer. It is based on attention mechanisms entirely without recurrence and convolutions. Experiments on machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to be trained. Their model achieves state-of-the-art performance on the machine translation task. They also show that the Transformer generalizes well to other sequence problem with large or limited training data.

3. Model Architecture

Most image captioning models have the encoder-decoder structure. The encoder is the CNN which maps an image to a context feature I. Given I, the decoder can generate an output sequence $(y_1, ..., y_m)$. The proposed model also contains two components, the encoder and the decoder, as is shown in Figure 1. The encoder is the CNN model, and the decoder is the Transformer model. At each step, the decoder is auto-regressive, consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers.

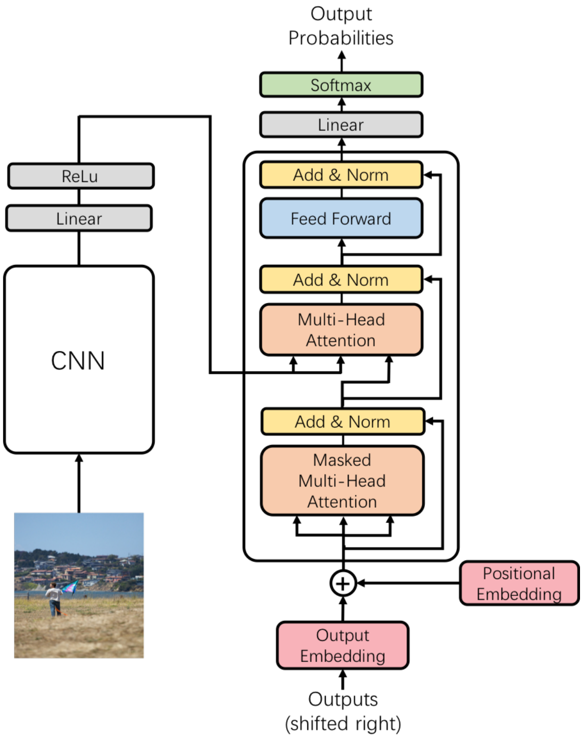


Figure 1. The framework of the proposed model. As shown in the figure, the model contains two components, the encoder and the decoder. The encoder is the CNN and the decoder is the Transformer. The CNN can use different state-of-the-art image classification models, such as ResNet and ResNext. The decoder is the Transformer with stacked Multi-Attention and Feed Forward Layer. In this figure, we only show one Transformer decoder layer. In practice, the Transformer decoder model contains N identical decoder layers.

3.1. Encoder and Decoder

3.1.1. Encoder

Different from the original Transformer model, we use the CNN as our encoder to extract image information. Original machine translation task is a sequence-to-sequence task. The Transformer model can encode the sequence input into a context vector, then this context vector is translated to the output sequence. However, the image caption task is to translate an image to a sequence. Therefore, we need to recognize the meaning of the image. In the recent proposed image caption models, the

CNN is successfully used to extract the meaning of the image. In Soft-Attention model [2], it can attend the salient objects in the image along the sentence generation. In practice, we use the ResNext [10] network to extract the image feature which is used in the image classification task. Higher layer of CNN usually has high-level semantic information of the image and it can get the global image information. To extract the spatial semantic information, we use the final convolution layer of the network. In the section Embedding, we will show how to embed the image into the image vector. For the high-level spatial information of the image, we add an adaptive pooling layer after the final convolution layer to get the wanted feature map size.

3.1.2. Decoder

The decoder is the Transformer model with stacked attention mechanisms, and it can decode the image feature into the sentence. The Transformer model does not contain any RNN structure. It composes of a stack of N identical layers, and each layer has three sub-layers. The first layer uses the multi-head self-attention mechanism. Figure 1 shows that the inputs of this layer are identical. This layer has a masked mechanism for preventing this model from seeing the future information. This masked mechanism can ensure the model generates the current word with only the previous words. The second layer is a multi-head attention layer without the masked mechanism. It performs the multi-head attention over the output of the first layer. This layer is the core layer to correlate the text information and the image information with the attention mechanism. The third layer is a simple, position-wise fully connected feed-forward network. The Transformer performs a residual connection around each of the three sub-layers, followed by layer normalization.

At the top, we add a full connected layer and a softmax layer to project the output of the Transformer to the probabilities for the whole sentence. Different from the LSTM, all the words in the sentence can be parallelly generated.

3.2. Embedding

3.2.1. Image Embedding

The CNN is used to encode the given image I to the spatial image feature. In practice, an image feature I is obtained from the pool-5 layer of the ResNext network [10]. The ResNext network is pre-trained on the ImageNet dataset [11]. We then apply adaptive-pooling, full connected linear and ReLU to obtain a d_{model}-dimensional image semantic feature and a image spatial feature $V = \{V_1, ..., V_{k\times k}\}$, $V_i \in R^d_{model}$, where $k \times k$ is the number of regions, and V_i represents a region of the image. This is consistent with the image feature used in the baseline Soft-Attention model [2].

3.2.2. Text Embedding

Firstly, all the words in the caption sentence are counted. If the number of word in all sentences is less than 5, we use the <UNK> token to replace with this word.With these reserved words, the dictionary is constructed, then we use this dictionary to represent each word. At last, we get the one-hot vector x for each word. The embedding model embeds the one-hot vector into a d_{model}-dimensional vector. All these embedding vectors in one sentence are combined into a matrix $L \times d_{model}$ as the input to the Transformer, where L is the length of the sentence.

3.3. Image Combination

We have tried three methods to combine the image feature to the Transformer model. First, we only use the image spatial feature map as the input of the second sub-layer of the Transformer. Second, we use the spatial image feature map as the input of the second sub-layer. Meanwhile we combine the image feature with each word embedding. Third, we use the spatial image feature map as the input of the second sub-layer and use the image feature before the start of the text embedding, as described in Neural Image Caption (NIC) [1]. We implement these three combination methods to find the best combination mechanism.In Section 4, we will make comparison experiments.

3.4. Attention in Transformer

3.4.1. Scaled Dot-Product Attention

The Transformer uses the Scaled Dot-Product Attention. The input consists of three input, i.e., keys K, values V and queries Q. This attention is shown as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where $Q \in R^{L \times d_k}$ is the query, $K \in R^{L \times d_k}$ is the key, $V \in R^{L \times d_v}$ is the value and the L is the length of the sequence. We can find that in Equation (1), $QK^T \in R^{L \times L}$ is the product operation, this could make the result become too large or small. This will influence the precision of the variable. So $\sqrt{d_k}$ is used to scale QK^T. At last, we get $Attention(Q, K, V) \in R^{L \times d_v}$. The dot-product attention is faster and space-efficient, and it can be implemented by the optimized matrix multiplication method. K and V is the key-value pair. If this pair at the first sub-layer of the Transformer, it will be the text embedding matrix. If this pair at the second sub-layer, it will be the image embedding matrix.

3.4.2. Multi-Head Attention

For better performing attentions, the multi-head attention is composed of n scaled dot-product attentions. The multi-head attention is shown as follows:

$$h_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

$$H = Concat(h_1, ..., h_n) \tag{3}$$

$$O = HW_h \tag{4}$$

where the projections are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$. $Q \in R^{L \times d_{model}}$, $K \in R^{L \times d_{model}}$, $V \in R^{L \times d_{model}}$ are the inputs of the multi-head attention. $Attention$ is the scaled dot-product attention, $Concat$ is the concat function. $h_i \in R^{L \times d_v}$ is the output of the scaled dot-product attention. n scaled dot-product attentions are concatenated to generate $H \in R^{L \times (n \times d_v)}$. We use $W_h \in R^{(n \times d_v) \times d_{model}}$ to project H into the output $O \in R^{L \times d_{model}}$.

In practice, different from the machine translation task, we need to combine the image with the text information. At the first decoder layer, all the inputs are all identical. That means that the keys, values and queries are the same matrices, and this mechanism is called the Self-Attention. This controls the relationship between the whole sequence. At the second decoder layer, the keys and the values are the matrices generated by the CNN, which reserves the spatial image information. The queries are the outputs by the first decoder layer, which means the sentence information. The target of this attention is to make the relation between the spatial information of the image and the sentence information.

3.5. Multi-Level Supervision

As is shown in Figure 2, we introduce a new multi-level supervision mechanism to leverage multi-layer outputs of the Transformer to generate the current word. Every layer of the Transformer can be used to generate the current word. At the inference, we use the average pooling layer to combine all the outputs to get the word probability. To train the model, we use the multi-output cross entropy loss. With these losses, every layer of the Transformer has the ability to learn the word information. Every layer can generate the current word, and the final result also benefits from different information of every layer.

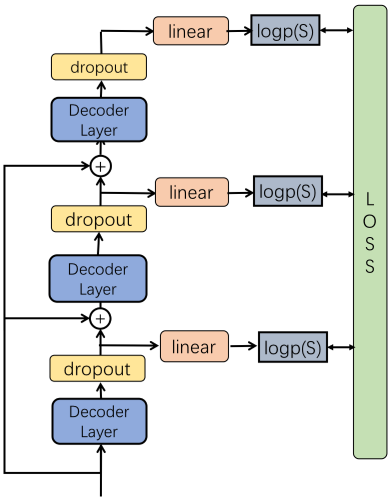


Figure 2. This figure shows the structure of the proposed multi-level supervision. We add a linear layer for every output, this makes the Transformer can generate sentence at every layer. At the training time, we train the model with all outputs.

From Figure 2, we can find that our model stacks three layers. The output of the first layer is combined with the second layer. Every layer we add dropout into the internal state of the Transformer. The fully connected layer is used to project the output of the transformer to the size L. Then we use the log softmax to get the current word probability. The output of the second layer uses the middle internal state of the decoder layer. All the three layers have the ability to generate the whole sentence. With the combination of three outputs, our model has the ensemble ability itself, and all the three outputs have the ability to generate the whole sentence. The standard Transformer uses the top output of the decoder layer. However, the bottom decoder layer only uses its output as the input of the upper decoder layer. The internal decoder layers cannot leverage the language information. The language information is only related with the top of the decoder layer, and this only increase the number of the parameters.

For not over-fitting, we add a dropout layer at every output of the decoder layer. We also use the residual mechanism, to accelerate convergence speed. The input of the upper decoder layer combines the word embedding with the output of the bottom layer. The number of the multi-level is also hyperparameter which can be set.

3.6. Training

The LSTM depends on the previous hidden state to generate current output. Different from the LSTM, the Transformer structure can be trained in parallel. The Transformer contains only the attention and feed-forward modules. It can be trained by one forward like CNN, and this will take full advantage of GPU. The total process can be described as follows. Firstly, an image will be sent to the CNN model, then we will get the image feature which has the same dimension as the word embedding vector. The CNN has two outputs. One is the spatial matrix and the other is the semantic vector. The spatial matrix is sent to the Transformer as the second sub-layer input. The semantic vector combines with the ground-truth sentence embedding matrix as the Transformer input. At the last, the model gets the probability distribution $p(S'|S, I)$ for the image, where I stand for the image feature, $S \in R^{L\times d_{model}}$ is the L length ground-truth sentence embedding matrix and $S' \in R^{L\times d_{model}}$ is the sentence generated by the model which shift right relative to the S. The Transformer now gets the whole sentence probability for the current image. To learn this model, we use the supervised learning method. Given the target ground truth sequence $S' = \{y_0, y_1, ..., y_t\}$, the model would be trained by minimizing the cross-entropy loss (XE) which is the same as that described in the NIC model [1]. It is shown as follows

$$logp(S|I) \;=\; \sum_{t=0}^{N} logp(S_t|I, S_0, ..., S_{t-1};\theta) \tag{5}$$

where θ is the parameter of the model, I is an image, S is the ground-truth sentence and (S, I) is the training example pair, We optimize the sum of the log probabilities as described in the above over the whole training set. We use the stochastic gradient descent method to train our model.

3.7. Inference

The inference is similar to the LSTM, and the word will be generated one by one at a time. Firstly, we also need to begin with the start token $< BOS >$, and generate the first word by $p(y_1|\phi, I)$. Afterwards, we get the dictionary probability $y_1 \sim p(y_1|\phi, I)$ at the first time. We can use the greedy method or the beam search method to select the possible word. Then, y_1 is fed back into the network to generate the following word y_2. This process will be continued until the end token $< EOS >$ is reached, or reach the max length L.

4. Experiments

For evaluating the proposed model, the MSCOCO dataset [12] is used. The MSCOCO dataset contains 82,783 training images, 40,504 validation images and 40,775 testing images. For comparing with other state-of-the-art methods, we use the same dataset splits as in [4]. Nowadays, this split is used as the offline evaluation. This training set of this split contains 113,287 images, 5000 validation images and 5000 testing images. The result of the model is reported on the testing dataset.

We use CIDEr [13], BLEU [14], METEOR [15] and ROUGE_L metrics to evaluate the quality of the generated sentences. CIDEr [13] measures consensus in image caption by performing a Term Frequency-Inverse Document Frequency weighting for each n-gram. BLEU [14] is a precision-based metric and it is traditionally used in the machine translation to measure the similarity between the generated captions and ground truth captions. METEOR [15] is based on explicit word to word matches between the generated captions and the ground-truth captions.

For the Transformer model, we set the model size which is d_{model} reported in Section 3.4.2 to be 512 and the mini-batch to be 16. We use the Adam method [16] to update the parameters of CNN and the Transformer. The initial learning rate of the Transformer is 4×10^{-4}, and the initial learning rate of the CNN is 1×10^{-5}. The momentum and the weight-decay are 0.8 and 0.999 respectively. We utilize the PyTorch deep learning framework to implement our algorithm. In inference, the beam search algorithm is used for better caption generation. In practice, we set the beam size to be 2.

We adopt the training strategy similar to NIC for the fair comparison. In the process of training Transformer network, we also use the CNN fine-tuning strategy proposed in NIC [1].

The NIC model and the Soft-Attention model which are based on the LSTM, are our baseline models. In order to maintain the fairness of the comparison, All the models use the same CNN model and hyperparameters. In practice, the number of the Transformer decoder layers is set to be 6, and the attention size used in the model is set to be 4×4.

5. Results

In order to further verify the performance of the proposed model, we conduct several comparative experiments with the state-the-art-of methods. In order to maintain the fairness of comparison, we also train the NIC model and the Soft-Attention, and we also use the same CNN as the encoder. In addition, all the super parameters in the training process keep the same. For evaluating the Multi-Level Supervision (MS) method, we also conduct several comparison experiments with the same super parameters but without CNN fine-tuning.

In Table 1, we present the performances of recently state-the-art-of methods. The NIC (Resnext_101_64×4d) model and the Soft-Attention (Resnext_101_64×4d) model [2] are our baseline models, and they are implemented by us with the same CNN as our CT model. They have better performance than the model reported in the original paper. The CT-C1-64a4n6, CT-C2-64a4n6, CT-64a4n6 models are our Caption Transformer (CT) models with different image combination methods where the CNN model is Resnext_101_64×4d. In practice, the number of decoder layers is set to be 6, and the attention size is set to be 4 × 4. The CT-C1-64a4n6 uses the first image combination method described in Section 3.3. The CT-C2-64a4n6 uses the second combination method which only uses the spatial image information at the second sub-layer. The CT-C3-64a4n6 uses the third combination method, which the input to the Transformer is similarly used in the NIC, and the spatial image matrices are used as the input of the second sub-layer of the Transformer. From Table 1, we can find that Image feature which combines with the text embedding as the input gets better performance than the two other methods. This shows that the third combination method can help the CT model combine with the image information better. Compared with other state-of-the-art methods, the CT-C3-64a4n6 model gets better performance than the Soft-Attention model and the NIC model on BLEU-3, BLEU-4, METEOR, ROUGE, CIDEr metrics. This means that the Transformer model can get better performance than LSTM. The BLEU-1, BLEU-2 scores are less than the NIC, but the margin is small. This shows that the Transformer model can be used in the image captioning task successfully, and has competitive result relative to the LSTM.

Table 1. The results of the Caption Transformer (CT) model compared with several state-of-the-art methods on standard evaluation metrics: BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE, CIDEr. CT-C1-64a4n6,CT-C2-64a4n6,CT-C3-64a4n6 are our proposed models, but with different combination method.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Google NIC [1]	-	-	-	27.7	-	23.7	85.5
Soft-Attention [2]	70.7	49.2	34.4	24.3	23.9	-	-
Hard-Attention [2]	71.8	50.4	35.7	25.0	23.0	-	-
VAE [17]	72.0	52.0	37.0	28.0	24.0	-	90.0
Google NICv2 [1]	-	-	-	32.1	25.7	-	99.8
Attributes-CNN + RNN [7]	74.0	56.0	42.0	31.0	26.0	-	94.0
CNN$_L$ + RHN [18]	72.3	55.3	41.3	30.6	25.2	-	98.9
PG-SPIDEr-TAG [19]	75.4	59.1	44.5	33.2	25.7	55.0	101.3
Adaptive [20]	74.2	58.0	43.9	33.2	26.6	-	108.5
NIC (Resnext_101_64×4d)	72.4	55.6	41.8	31.4	-	53.7	100.9
Soft-Attention (Resnext_101_64×4d)	**73.7**	**57.1**	43.3	32.6	-	-	104.6
CT-C1-64a4n6 (Resnext_101_64×4d)	71.8	55.4	41.8	31.5	-	54.7	105.8
CT-C2-64a4n6 (Resnext_101_64×4d)	73.3	57.0	43.6	33.2	-	**55.1**	107.0
CT-C3-64a4n6 (Resnext_101_64×4d)	73.0	56.9	**43.6**	**33.3**	-	54.8	**108.1**

In Table 2, we evaluate the proposed multi-level supervision method on BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L, CIDEr metrics. The CT-ms-o1-64a4n6 model is our baseline model, and it is the same as the standard CT model. All the six models are trained without the CNN fine-tuning for saving time. The CT-ms-on-64a4n6 model means that we ensemble the top n decoder layers to generate the current output. We can find that with different outputs ensemble, these models show different performances. When we ensemble 6 decoder results, the model gets the best result on the evaluation metrics. It means that the proposed Multi-Level Supervision (MS) method can achieve better performance and it is useful for the training of the CT model.

In Table 3, we report the results about the effect of the different decoder layers. We use three different decoder layers to find the appropriate settings. The CT-C3-64a4nm model means that we use m decoder layers as the decoder. All the three models use the third combination method, the encoder is Resnext_101_64×4d and the attention size is set to be 4 × 4. From these results, we find that when the decoder layer is setting 6, the model gets the best performance. It means that the more decoder layer can make the model achieve better performance, but the decoder will have more parameters, will make the model become bigger and train slower.

Table 2. The results of the Multi-Level Supervision method with different super parameters.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr
CT-ms-o1-64a4n6	69.9	52.8	39.4	29.6	52.5	94.9
CT-ms-o2-64a4n6	69.8	52.9	39.5	29.6	52.4	95.0
CT-ms-o3-64a4n6	70.4	53.3	39.8	29.8	52.7	95.8
CT-ms-o4-64a4n6	70.6	53.5	40.1	30.3	52.9	96.8
CT-ms-o5-64a4n6	70.0	52.8	39.4	29.6	52.3	95.6
CT-ms-o6-64a4n6	70.9	54.2	40.9	31.0	53.2	98.7

Table 3. The results of the CT model with the different decoder layers.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr
CT-C3-64a4n2	72.6	56.1	42.6	32.3	54.5	104.4
CT-C3-64a4n4	72.5	55.9	42.4	32.2	54.6	105.2
CT-C3-64a4n6	73.0	56.9	43.6	33.3	54.8	108.1

In Table 4, we report the results of the different CNN usage. We use the Resnet_152 as the encoder to evaluate the effect of different CNNs. We use the same training settings as the models reported in Table 1. The performances of the models in Table 4 are slightly worse than models in Table 1, but our CT model can also achieve better performance than original NIC and Soft-Attention model under the same CNN.

Table 4. The results of the CT model with the Resnet_152 as the encoder.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr
NIC (Resnet_152)	71.7	54.8	41.0	30.8	53.2	98.8
Soft-Attention (Resnet_152)	73.4	56.9	43.1	32.6	54.4	103.5
CT-C3-64a4n6 (Resnet_152)	72.9	56.7	43.3	33.1	54.9	107.8

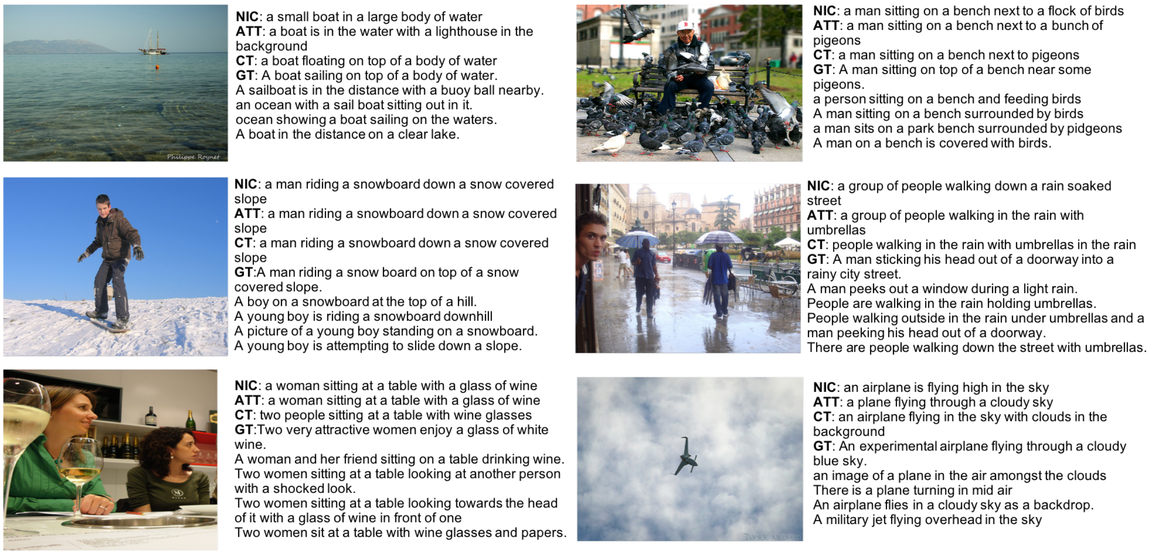
From Table 5, we can find that the CT model is faster per parameter than the LSTM. It is because that the CT model can be trained in parallel and do not have the timing dependence structure.

Table 5. Comparison of training time per batch for our CT model and the Soft-Attention model. The timings are obtained on single Nvidia Titan X Graphics Processing Unit (GPU).

Methods	Batch Size	Parameters	Time
CT-C3-64a4n6	16	27.5 M	0.245 s
Soft-Attention	16	14.3 M	0.177 s

6. Quantitative Analysis

As is shown in Figure 3, we select some samples from the local test set for reference. ATT is the Soft-Attention model, CT is our model, and GT is the ground-truth which has five sentences. We can see that our model can generate readable text content and maintain rich semantic information about the image. 5 please define. For example, in the first image, we can see the generated text "a boat floating on top of a body of water". The generated caption can successfully describe the boat and the water in the image, and even is nearly the same as the first ground-truth sentence. In the second image, we can see our model can recognize man, bench and pigeons in the image, and even more, it can find that a man is sitting on the bench.



NIC: a small boat in a large body of water
ATT: a boat is in the water with a lighthouse in the background
CT: a boat floating on top of a body of water
GT: A boat sailing on top of a body of water.
A sailboat is in the distance with a buoy ball nearby.
an ocean with a sail boat sitting out in it.
ocean showing a boat sailing on the waters.
A boat in the distance on a clear lake.

NIC: a man sitting on a bench next to a flock of birds
ATT: a man sitting on a bench next to a bunch of pigeons
CT: a man sitting on a bench next to pigeons
GT: A man sitting on top of a bench near some pigeons.
a person sitting on a bench and feeding birds
A man sitting on a bench surrounded by birds
a man sits on a park bench surrounded by pidgeons
A man on a bench is covered with birds.

NIC: a man riding a snowboard down a snow covered slope
ATT: a man riding a snowboard down a snow covered slope
CT: a man riding a snowboard down a snow covered slope
GT:A man riding a snow board on top of a snow covered slope.
A boy on a snowboard at the top of a hill.
A young boy is riding a snowboard downhill
A picture of a young boy standing on a snowboard.
A young boy is attempting to slide down a slope.

NIC: a group of people walking down a rain soaked street
ATT: a group of people walking in the rain with umbrellas
CT: people walking in the rain with umbrellas in the rain
GT: A man sticking his head out of a doorway into a rainy city street.
A man peeks out a window during a light rain.
People are walking in the rain holding umbrellas.
People walking outside in the rain under umbrellas and a man peeking his head out of a doorway.
There are people walking down the street with umbrellas.

NIC: a woman sitting at a table with a glass of wine
ATT: a woman sitting at a table with a glass of wine
CT: two people sitting at a table with wine glasses
GT:Two very attractive women enjoy a glass of white wine.
A woman and her friend sitting on a table drinking wine.
Two women sitting at a table looking at another person with a shocked look.
Two women sitting at a table looking towards the head of it with a glass of wine in front of one
Two women sit at a table with wine glasses and papers.

NIC: an airplane is flying high in the sky
ATT: a plane flying through a cloudy sky
CT: an airplane flying in the sky with clouds in the background
GT: An experimental airplane flying through a cloudy blue sky.
an image of a plane in the air amongst the clouds
There is a plane turning in mid air
An airplane flies in a cloudy sky as a backdrop.
A military jet flying overhead in the sky

Figure 3. Examples of the generated sentences by our model. This figure contains six images which are randomly selected from the validation set.

7. Conclusions

We present the new Caption Transformer model which does not rely on the RNN model. We only use stack attention layers to learn the sequence relationships among the language. This structure is inherent and it can be trained in parallel, without the timing dependency problem. Then we introduce the multi-level supervision training method to improve the model performance. With these innovations, we get the competitive performance on the MSCOCO benchmark. We intend to study the application of the proposed method in the field of digital virtual asset security in future research.

Author Contributions: X.Z., L.L., J.L. and H.P. conceived and designed the experiments; X.Z. performed the experiments; X.Z., L.L., J.L. and H.P. analyzed the data; X.Z., L.L. and J.L. wrote the paper. All authors interpreted the results and revised the paper.

Acknowledgments: This work is supported by the National Key R&D Program of China (Grant No. 2016YFB0800602), and the National Natural Science Foundation of China (Grant Nos. 61472045, 61573067).

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

References

1. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 652–663. [CrossRef] [PubMed]
2. Xu, K.; Ba, J.L.; Kiros, R.; Cho, K.; Courville, A.C.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Int. Conf. Mach. Learn.* **2015**, *3*, 2048–2057.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of NIPS 2017: The Thirty-first Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
4. Karpathy, A.; Feifei, L. Deep visual-semantic alignments for generating image descriptions. *Comput. Vis. Pattern Recognit.* **2015**, 3128–3137. [CrossRef]
5. Jia, X.; Gavves, E.; Fernando, B.; Tuytelaars, T. Guiding the Long-Short Term Memory Model for Image Caption Generation. *Int. Conf. Learn. Represent.* **2015**, 2407–2415. [CrossRef]
6. Yang, Z.; Yuan, Y.; Wu, Y.; Salakhutdinov, R.; Cohen, W.W. Encode, Review, and Decode: Reviewer Module for Caption Generation. *arXiv* **2016**, arXiv:1605.07912.
7. Wu, Q.; Shen, C.; Liu, L.; Dick, A.R.; Den Hengel, A.V. What Value Do Explicit High Level Concepts Have in Vision to Language Problems. *Comput. Vis. Pattern Recognit.* **2016**, 203–212. [CrossRef]

8. Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; Mei, T. Boosting Image Captioning with Attributes. In Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
9. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; Zhang, L. Bottom-up and top-down attention for image captioning and vqa. *arXiv* **2017**, arXiv:1707.07998.
10. Xie, S.; Girshick, R.B.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *Comput. Vis. Pattern Recognit.* **2016**, 1492–1500. [CrossRef]
11. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
12. Lin, T.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
13. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-based image description evaluation. *Comput. Sci.* **2015**, 4566–4575. [CrossRef]
14. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, Pennsylvania, 7–12 July 2002; pp. 311–318.
15. Denkowski, M.; Lavie, A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. *Workshop Stati. Mach. Transl.* **2014**, 6, 376–380.
16. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
17. Pu, Y.; Gan, Z.; Henao, R.; Yuan, X.; Li, C.; Stevens, A.; Carin, L. Variational Autoencoder for Deep Learning of Images, Labels and Captions. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2352–2360.
18. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, L.; Wang, G. Recent Advances in Convolutional Neural Networks. *Comput. Sci.* **2015**, arXiv:1512.07108.
19. Liu, S.; Zhu, Z.; Ye, N.; Guadarrama, S.; Murphy, K. Improved Image Captioning via Policy Gradient optimization of SPIDEr. *arXiv* **2016**, arXiv:1612.00370.
20. Lu, J.; Xiong, C.; Parikh, D.; Socher, R. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *arXiv* **2016**, arXiv:1612.01887.