Image Captioning : Exploring CNN + CNN And CNN + Transformer Model

Jiawei Guo
Department of Physics
Carnegie Mellon University
Pittsburgh, PA 15213
jiaweigu@andrew.cmu.edu

Xiaowen Zhang
Department of Physics
Carnegie Mellon University
Pittsburgh, PA 15213
xiaowen4@andrew.cmu.edu

Gen Li
Department of Physics
Carnegie Mellon University
Pittsburgh, PA 15213
genli@andrew.cmu.edu

Sitong An
Department of Physics
Carnegie Mellon University
Pittsburgh, PA 15213
sitonga@andrew.cmu.edu

Abstract

Image captioning is a process that automatically generates a description for an image. We explored two different approaches that make heavy use of the attention mechanism: CNN + CNN and CNN + Transformer, and compared their performances on the MSCOCO dataset. We analysed the effectiveness of the CNN + CNN approach, and demonstrated that the CNN + transformer has the potential to address the weakness identified in the former. We also discussed some of the practicalities of training transformer model.

1 Introduction

Image captioning is a process that automatically generates a description for an image. The task combines the techniques from both the fields of computer vision and natural language processing. The core challenge in this task is the requirement to recognize the main objects in the image, to comprehend the relationships between them, and to generate syntactically and semantically understandable sentences.

With the rapid development of Deep Learning in the last decade, image captioning has become one of the hottest topics due to its various applications. It can be used to do image indexing and classifying, which is important for content-based image retrieval. For example, NVIDIA is trying to use image captioning technologies to help people who have low or no eyesight. In addition, on social media platforms like Instagram or Twitter, it can be used to directly generate captions for a post.

2 Related Work

Many approaches have been demonstrated to achieve image captioning with deep learning techniques[5, 10, 4, 12, 1, 2]. Deep learning techniques have been shown to be capable to handling large image dataset, as well as natural languages such as in tasks like machine translation. Typically, the model consists of an encoder and a decoder. The encoder acquires features from image and sentences and encodes them in tensors in the representation space. The decoder generates captions by amalgamating information from both image and word features in the representation space and

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

predicting the words with highest probabilities. For the image model, most works use Convolutional Neural Network(CNN) as encoder, but for the language model there are different choices.

2.1 Recurrent Neural Network

The most popular method for language model is Recurrent Neural Network(RNN). Mao et al. proposed the m-RNN model to achieve image captioning[5]. The language RNN output is fed with the CNN image output into a multimodal block to establish connections between words and image but the model only predicts one word for every step, which means it can only learn the connections between the nearest words.

Then, Vinyals et al. proposed the neural image captioning(NIC) model to solve the problem[10]. The NIC model employs a long-short term memory net as a decoder, which can store information from several words. Therefore, the LSTM net has a high performance on sequence task. Furthermore, Jia et al. proposed gLSTM model[4].The model introduces the semantic information extracted from image to guide LSTM net to generate captions that couple with the image more tightly.

Later on, with the trend that attention mechanism is widely used in machine translation and object detection, Xu et al. presented an attention based model which can automatically capture image information[12]. Attention mechanism is invented by mimicking cognitive attention, which is able to enhance some parts of the input image and reduce others. As a consequence, the model is forced to focus on the main objects in the image and learns how to distinguish important parts from the trivials in the image to generate more precise captions.

Although the RNN method offers cheering results for image captioning, but it cannot be parallel programming, which is a constrain for high speed training. On the other hand, RNN loses long-range information for sentences so that it is difficult to learn grammars.

2.2 Convolutional Neural Network

CNN is an alternative method for language encoders, which is able to be computing parallel and handle longer sentences with multiple layers. Gu et al. introduced CNN language model that can feed with all history words to predict the next word[2]. The CNN language model is capable to learn long-range dependence between history words and shows competitive performance. Furthermore, Aneja et al. performed a detailed analysis on CNN language model and demonstrated its performance on par with LSTM but faster to training[1].

3 Data

3.1 MSCOCO Dataset

MSCOCO is the most popular dataset for image captioning, which includes more than 80 object categories and more than 200K labeled images. Here we use MSCOCO 2017, which consists of of four parts: train (118K examples), test (40K examples), validation (5K examples) and unlabeled (120K examples). For each image in train set, it has 5 different annotations corresponds, which can let our model learn more ways of expressions, and can avoid overfitting. By captioning unlabeled images with our model, we can have a more direct intuition about how well is the training.

3.2 Preprossessing

For our images, we follow a standard preprocesssing flow: First we use $RandomResizedCrop(224)$ to crop a random area of image with a random aspect ratio, then resize it to the given size 224. This is popularly used to train the Inception networks. Then we randomly flip the image and normalize it into $\text{mean} = [0.485, 0.456, 0.406],\ \text{std} = [0.229, 0.224, 0.225]$, where numbers correspond to R,G,B channels of the image.

For our captions, following a standard way, we remove all the punctuation, like , ; . :!?′, to make our word-embedding and training easier.

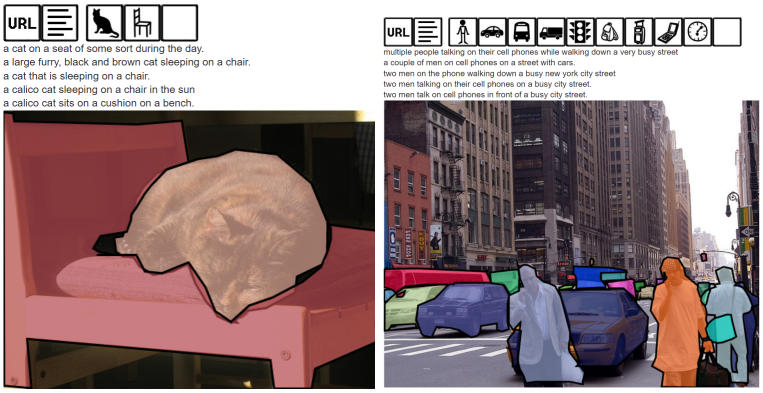


Figure 1: COCO dataset. As you can see, each image corresponds to 5 different captions. The complexity of images in COCO can vary hugely, some images (left) only contain a few objects, while the others (right) contain multiple objects, which can cause difficulties to our model.

4 Methods

The baseline of our project is using hierarchical CNN+CNN to realize our image captioning. For the image encoder, we utilized a pretrained CNN model, VGG-16, as our vision module. For the language module, we used attention based hierarchical CNN model, which includes language module and attention module as our baseline method. At the end, there is a prediction module to generate captions as output.

Then, we proposed to replace the language encoder with Transformers as a extension for our model and we compared the performance of the 2 different models in Section 6.

4.1 Vision Module

VGG-16 [8] is a pretrained version of the network trained on more than a million images, which can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. We use VGG-16 as our vision module to extract features from images, and with preprocessed 224×224 images as inputs and with $d \times d \times D_c$ feature maps as outputs, where D_C is the dimension of feature vector which represents part of the image. For the further discussion, we define $\mathbf{v} = [v_1, ..., v_N]$, where $N = d \times d$ and $v_i \in \mathrm{R}^{D_c}$, then we can use $\mathbf{v}$ to represent our vision module output for each image.

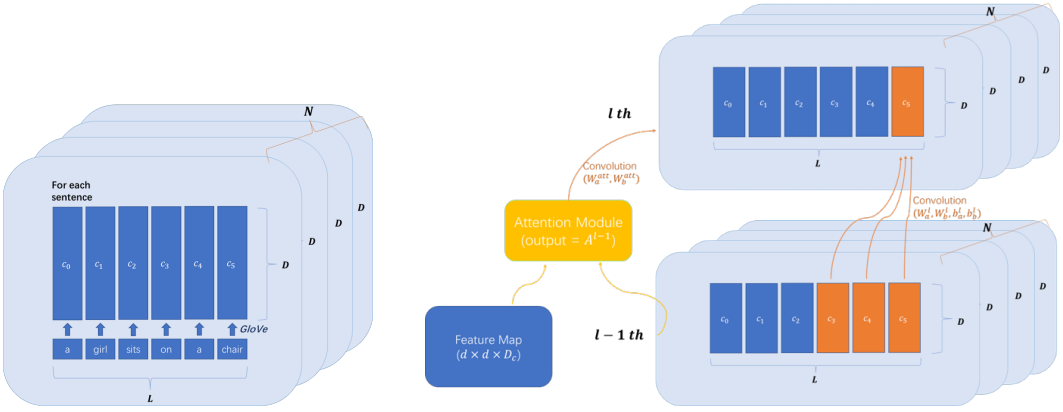
4.2 Word Embedding and Inputs

We use GloVe [7] (Global Vectors for Word Representation) to embed our words into vectors. We first divide our inputs, which are images with captions, into batches with batch-size $N = 10$. For each sentence in a batch, we use GloVe to embed each word in the sentence into $D = 300$ dimension vectors c_j, which can be called as concepts. Then for each batch, we find the longest sentence with length L_{max}, and we pad all the sentences in this batch into length L_{max} with zero vectors. Therefore, the output of embedding for each batch is a $N \times L_{max} \times D$ dimension tensor (Figure 2a), which will become the input of language module. For convenience of our further discussion, we define each sentence as $\mathbf{c} = [c_1, ..., c_L]$, $L = L_{max}$.

4.3 Hierarchical CNN+CNN Model

Inspired by CNNs used for NLP, we propose a hierarchical CNN+CNN frame work for image captioning as our baseline. In the Hierarchical CNN+CNN Model (Fig.2 (b)), the language module is integrated with vision module and attention module at each level of the hierarchical model. Like we have described in the previous section, feature maps that come from VGG-16 as our vision module's

outputs, together with the outputs of each convolution layer of language module, become the input of the attention module for each level of model.



(a) Input for Language Module (b) Hierarchical Model

Figure 2: (a)Input for language module: Coming from embedding, for each batch, it is a $N \times L_{max} \times D$ dimension tensor, where $N = 10$ is the batch-size, $L_{max}$is the length of longest sentences in each batch, and $D = 300$ is the embedding vector's dimension. (b)Hierarchical Model: We add an attention layer after each convolution layer of language model, and fed into next level. Thus different levels of the language CNN represent different concept levels could all benefit from visual inputs.

4.3.1 Attention Module

Just as shown in Figure 3, we take visual features $\{v_i\}$from vision module and concepts $\{c_j\}$ from language module as our attention module inputs. For each $v_i \in R^{D_c}$ and $c_j \in R^D$ we can calculate a score $s_{ij} = v_i^T U c_j$, where U is a $D_c \times D$ matrix. The score matrix combine the information from vision and language modules. Then we apply a softmax function to the score matrix s_{ij}, $w_{ij} = \frac{e^{s_{ij}}}{\sum_{i,j} e^{S_{ij}}}$, and get the weight matrix w_{ij}. Finally, we use the weighted sum operation to calculate the final attention vectors $a_j = \sum_{i,j} w_{ij} v_i$. Each element a_j in the attention vector A_l (l indicates the number of layer) corresponds to a word in the sentence. The attention module could let the CNN model focus more on the important information (objects in the image, and the grammar of captions), rather than learning non-useful background information.

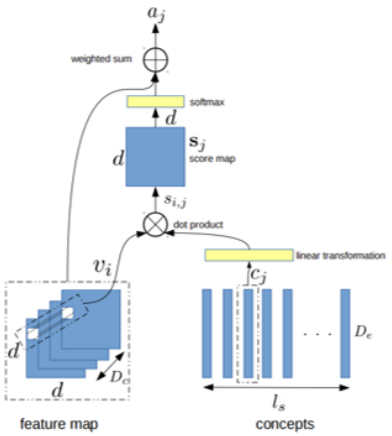


Figure 3: Attention Module[11]: Attention module takes feature maps that from vision module and concepts from each level of language module as inputs. The outputs are the attention features a_j, corresponding to each word in the sentence .

4.3.2 Hierarchical Language Module

Just as shown in Figure 2b, each level of hierarchical language CNN takes the concepts c_j and attention vector A_l as inputs. If we use $h_l = \mathbf{c} = [c_1, ..., c_L]$ to indicate the concepts matrix for each layer. The convolution relation between each level could be written as:

$$\begin{cases} h_a^l &= W_a^l * h^{l-1} + W_a^{att} * A^{l-1} + b_a^l \\ h_b^l &= W_b^l * h^{l-1} + W_b^{att} * A^{l-1} + b_b^l \\ h^l &= h_a^l \odot \sigma\left(h_b^l\right) \end{cases} \tag{1}$$

$W_a^l,\ W_b^l$ are kernels for language module of the lth layer, $W_a^{att},\ W_b^{att}$ are kernels for attention module, $b_a^l,\ b_b^l$ are biases, $*$ denotes the convolution operator, $\odot$ denotes the element-wise multiplication. $\sigma(x)$ is the sigmoid function. In general, h_b^l plays the role of a gate and h_a^l is a linear transformation.

Since VGG-16[8] and GloVe[7] are pretrained, the training parameters for our Hierarchical CNN model are $\{W_a^l,\ W_b^l,\ \forall l\}$, and $\{W_a^{att},\ W_b^{att}\}$.

4.4 Transformer Language Module

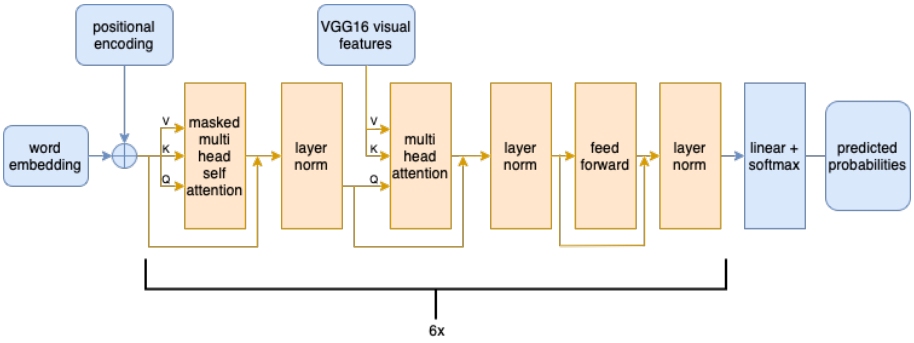


Figure 4: The transformer decoder architecture. Self attention allows the network to attend to information in embedding from other sentence positions. A mask is applied to prevent each word position from attending to subsequent positions. Residue connections are employed around each attention layer.

To extend the use of attention mechanism, we propose to replace the CNN language module with a transformer decoder[9]. There are two core ideas to this architecture: self attention and multi head attention. In self attention, the word embeddings are used as the value, the key as well as the queries. This allows the network to establish arbitrarily long-range connections between words far away from each other in a sentence with constant information path length. In other words, compared with the previous hierarchical language module which employs a CNN with kernel size 3, a self attention layer effectively has a kernel size of infinity. Multi head attention, on the other hand, allows the model to attend to information from different positions in its representations subspace.

During training, we used the warm-up and weight decay technique on the Adam optimizer, following [3]. Transformer training is prone to instabilities and these techniques proved essential.

5 Results

5.1 Metric

We use BLEU (Bilingual Evaluation Understudy)[6] score as our metric.

5.1.1 BLEU Score

BLEU score is between 0 and 1 which represent the similarity between the machine generated sentence(or translation) and high-quality reference sentence. Below is the interpretation of BLEU scores.

Table 1: BLEU score and its interpretation

BLEU Score	Interpretaion
< 10	Almost useless
$10 - 19$	Hard to get the gist
$20 - 29$	The gist is clear, but has significant grammatical errors
$30 - 40$	Understandable to good translations
$40 - 50$	High quality translations
$50 - 60$	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

The model reaches good representation of current caption when the score is higher than 0.6 (60%).

5.1.2 BLEU Score details

The BLEU score is calculated based on sentence to sentences similarity. Given the reference from the standard translation and the generated sentence (candidate) from neural network. Say the length of sentence is n and there are m words in the reference sentence, the BLEU-1 score is just $\frac{m}{n}$. The equation of BLEU score is below:

$$\text{BLEU}_n = \frac{\sum_{c \in \text{candidates}} \sum_{\text{n-gram} \in c} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{c' \in \text{candidates}} \sum_{\text{n-gram} \in c'} \text{Count(n-gram')}} \tag{2}$$

The first sum in numerator is all the candidate and second is all the n-gram in one candidate. "Count" refer to the number of n-gram in reference. So the whole numerator is how many n-gram present in reference given candidate. The denominator is the total number of n-gram in all candidate. Since here we count the score of matching words, the more reference sentence we have, the more accurate we can measure the quality of translated sentence.

Next, we calculate the the geometric mean of the n-gram score. Besides, we also have brevity penalty on machine generated sentence. Assume the length of reference sentence is r and the length of candidate sentence is c, our penalty is :

$$\text{Penalty} = \begin{cases} 1 & \text{if} \quad c > r \\ e^{1-r/c} & \text{if} \quad c < r \end{cases} \tag{3}$$

Then our BLEU score is:

$$\textbf{BLEU} = \text{Penalty} * \exp \sum_n w_n \log(\text{BLEU}_n) \tag{4}$$

5.2 CNN+CNN Results

Figure 5 shows the result for CNN-CNN baseline model, which is trained on 110k data and all 5 annotation for each image. Here, the generated captions are for images from the validation set. We can see that the first output of caption matches almost perfectly with the content in the image. However, in the second image, it fails to capture multiple objects. The output after the "." is because we padded the sentence to same length while training and it was not counted while calculating the loss. So the words after first ".", i.e. end of sentence, is not counted as valid output.

As we can see, our caption for the image always start with "a" instead of other words. We think it's because our model capture the main character in the image first, so during prediction the main component of the image will have the highest probability. Also the model fails to produce a complete sentence with multiple components, it will only produce each word to get higher score.

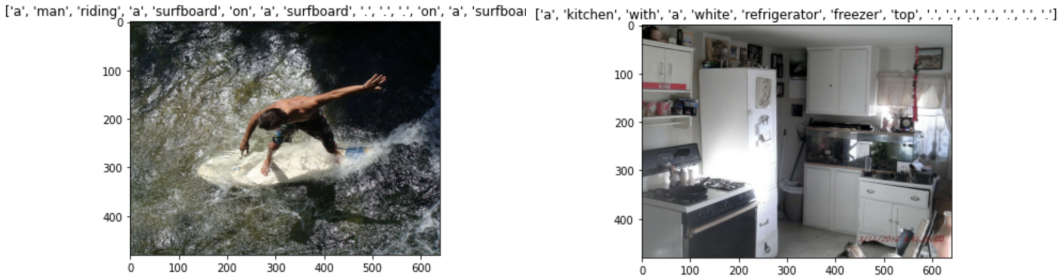


Figure 5: Examples of CNN+CNN model on validation dataset. The words above the image are the generated caption for the corresponding image.

Table 2: BLEU score, Each model is trained with label caption randomization.

Model	Train	Validation			
	BLEU-1	BLEU-1	BLEU-2	BLEU-3	BLEU-4
CNN + CNN (10k data)	0.63	0.38	0.15	0.04	0.01
CNN + CNN (110k data)	0.52	0.51	0.31	0.14	0.05
CNN + Transformer (10k data)	0.70	0.40	0.18	0.06	0.02

5.3 Transformer Results

The training of transformer model turned out to be challenging and we applied learning rates warm-up and decay to Adam optimizer in order to stabilize the training process. The model is trained on 100 batches of batch size 100, for a total of 10000 training samples.

Figure 6: Examples of validation output from the transformer model.

6 Discussion and Analysis

6.1 CNN+CNN Model

We found that most of our captions start with "a" + the main component, which is not the case in training label. The main reason to this might because that during the prediction process, the model will predict the most possible word first, it makes sense that the main component of the image will have the highest probability. Although the loss for this caption will not be very small because of the negative log-likelihood loss. Using a differentiable BLEU-like loss function may help improve the model more.

We also find that the simultaneous presence of multiple subjects in a photo poses challenge for the CNN+CNN model. This is likely due to the fact that the attention module used in the hierarchical

CNN model is single-headed. Averaging and normalization of the attention weights prohibits attention to multiple positions in the representation subspace simultaneously.

6.2 Transformer Model

A quantitative comparison of the BLEU scores across different models is in Table 2. At the same dataset size of 10k, transformer is able to outperform our baseline CNN+CNN model on validation BLEU score. Indeed we have seen indications that the transformer model has improved upon areas of potential improvement that we have identified for our baseline CNN+CNN model. For example, the generated caption for the image on the right of Figure 6 shows that the model is able to simultaneously attend to the colour of the clothes of the main subject of the photo, as well as related objects interacting with it. This is possibly a demonstration of the effect of the multi-head attention layer.

However we note that transformer model does suffer from over-fitting to the small training set due to the constraint of the timescale of this project. The example on the left of Figure 6 demonstrates this, where the network successfully identifies the main subject of the photo (scissors) but presumably overfits to the caption of an sample in the training set. This overfitting can also be seen in the large difference in the training and validation BLEU scores.

We also note the significant improvement of validation BLEU scores for the baseline CNN+CNN model when we scale up our training set from 10k to 110k as seen in Table 2. Given enough time, if the transformer model is trained on a dataset of the same size, it would likely have better performance.

In our original proposal we surmised that transformers might enjoy higher training speed due to potential parallelization. In practice, due to the restrictions imposed on the learning rate essential for the training of transformers, we did not find the training of transformers to be less time or compute intensive.

References

[1] Jyoti Aneja, Aditya Deshpande, and Alexander G. Schwing. "Convolutional Image Captioning". English (US). In: *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Publisher Copyright: © 2018 IEEE.; 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018 ; Conference date: 18-06-2018 Through 22-06-2018. IEEE Computer Society, Dec. 2018, pp. 5561–5570. DOI: 10.1109/CVPR.2018.00583.

[2] Jiuxiang Gu, Gang Wang, and Tsuhan Chen. "Recurrent Highway Networks with Language CNN for Image Captioning". In: *CoRR* abs/1612.07086 (2016). arXiv: 1612.07086. URL: http://arxiv.org/abs/1612.07086.

[3] Austin Huang et al. *The Annotated Transformer*. 2022. URL: http://harvardnlp.github.io/annotated-transformer.

[4] Xu Jia et al. "Guiding Long-Short Term Memory for Image Caption Generation". In: *CoRR* abs/1509.04942 (2015). arXiv: 1509.04942. URL: http://arxiv.org/abs/1509.04942.

[5] Junhua Mao et al. "Explain Images with Multimodal Recurrent Neural Networks". In: *CoRR* abs/1410.1090 (2014). arXiv: 1410.1090. URL: http://arxiv.org/abs/1410.1090.

[6] Kishore Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: 2002, pp. 311–318.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[8] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

[9] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[10] Oriol Vinyals et al. "Show and Tell: A Neural Image Caption Generator". In: *CoRR* abs/1411.4555 (2014). arXiv: 1411.4555. URL: http://arxiv.org/abs/1411.4555.

[11] Qingzhong Wang and Antoni B. Chan. "CNN+CNN: Convolutional Decoders for Image Captioning". In: *CoRR* abs/1805.09019 (2018). arXiv: 1805.09019. URL: http://arxiv.org/abs/1805.09019.

[12] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2048–2057. URL: https://proceedings.mlr.press/v37/xuc15.html.