Image Captioning and Comparison of Different Encoders

Ankit Pal¹, Subasish Kar¹, Anuveksh Taneja¹, Vinay Kumar Jadoun²

¹National Institute of Technology Delhi, India
²Manipal Institute of Technology, MAHE, Manipal, Karnataka, India

E-mail: ankapal@ieee.org, subasishkar35@gmail.com, anuveksh.taneja@gmail.com, vjadounmnit@gmail.com

Abstract. Generation of a sentence given an image, called image captioning, has been one of the most intriguing topics in computer vision. It incorporates knowledge of both image processing and natural language processing. Most of the current approaches integrates the concepts of neural network. Different predefined convolutional neural network (CNN) models are used for extracting features from an image and uni-directional or bi-directional recurrent neural network (RNN) for language modelling. This paper discusses about the commonly used models that are used as image encoder, such as Inception-V3, VGG19, VGG16 and InceptionResNetV2 while using the uni-directional LSTMs for the text generation. Further, the comparative analysis of the result has been obtained using the Bilingual Evaluation Understudy (BLEU) score on the Flickr8k dataset.

1. Introduction

An image contains a lot of information. With a single glance, a human eye can decipher that information without much difficulty. Automated image captioning aims to do the same; it generates description of a given image, that is, it extracts the information from a given image and converts it into a language that can be easily understood by any person. This task is helpful in many real-life applications like aid for the visually disabled, in case of security surveillance and many more.

Many different approaches have been proposed in deep learning for, extracting features of an image, and natural language processing [1][2]. Generally convolutional neural network models are used as the encoder to extract features from the images; moreover, the recurrent neural network is used as decoder, to decode the extracted features into a description [2][3].

There can be many possible captions for a single image, and different models can give different captions. This paper compares the encoding models on the basis of BLEU score on the same dataset and language model.

There are many datasets available for the English image captioning task, such as Flickr30k [4], Flickr8k [5] and MS-COCO [6]. This paper uses Flickr8k dataset, that has all different kinds of images.

2. Related Work

2.1. Past Work

This section gives the relevant background on image caption generation. There have been many approaches regarding the image captioning task, [1] proposed a method of learning the mapping between images and captions using a graphical model using features engineered by humans. Whereas [7] and [8] uses the modern convolutional neural networks for encoding and recurrent neural networks for language modelling, this is end to end training.

The other approach followed in [9] and [10], first train a convolutional and bi-directional recurrent neural network that try to learn the mapping of images and fragments of captions to the same embedding. Then it uses a different recurrent neural network that learns to combine the inputs from various object fragments detected in the original image to form captions.
Further improvement in the working of the above approaches can be seen by incorporating attention mechanism. Attention allows the model to focus on specific part of the input while neglecting others. This has been addressed in [3] and the variational auto-encoders for addressing this task in [11].

2.2. Dataset
Flickr8k dataset is provided by The Department of Computer Science in University of Illinois at Arbana-Champaign [2]. It contains 8k images, 6k training images, 1k validation images and 1k testing images.

![Figure 1. Sample image of dataset](image)

The sample image from the dataset is shown in Fig 1, and its corresponding five captions are as follows:

(i) A brown Dog is sprayed with water.
(ii) A dog is being squirted with water in the face outdoors.
(iii) A dog stand on his hind feet and catches a stream of water.
(iv) A jug is jumping up it is squirted with a jet of water.
(v) A tan, male dog is jumping up to get a drink of water from a spraying bottle.

3. Image Captioning
The aim of this paper is to compare the working of different convolutional models and obtain the result using the BLEU metric. Images are encoded into a vector of features using different models. Further, these features are used in the common decoding model.

3.1. Language Processing
For the language processing task, a map is created in the form of a data-frame, where each image has five captions, as 0,1,2,3 and 4. The words that occur frequently are deleted from the caption before feeding it to the network, so that the test set gives a more general description and does not overfit any word.
3.2. Encoders: Convolutional Models

In this section, four encoders are discussed. Descriptions of these encoders are written in 3 subsections. Subsection 1, 2 and 3 discuss about encoder VGG16 and VGG19, Inception-V3 and Inception-ResNet-V2, respectively.

3.2.1. VGG16 and VGG19: VGG16 and VGG19 [12] takes a fixed size input of 224 x 224 RGB image. Image is passed through the different layers of convolutional network with the kernel size of 3 x 3. Convolutional layers are followed by three fully connected layers (first two have 4096 channels and the third has 1000 channels). The main difference in these layers is that VGG16 has 13 convolutional layers and VGG19 has 16 convolutional layers. Architectural layers of VGG16 and VGG19 are shown in Fig 2 and Fig 3, respectively.

![Figure 2. VGG16 architecture layers](image)

3.2.2. Inception-V3: Information available in an image can vary in location and size, so picking suitable kernels is difficult for different images. Inception-V3[13] model uses different kernels of size 1x1, 3x3, 5x5. Inception-v3 takes a fixed input of 299 x 299 RGB image, and it uses batch normalization in the auxiliary classes. The total depth is the sum of the depth of each layer of kernels. Last layer of the network is a linear layer of size 1x1x1000. The architectural layers of Inception-V3 is shown in Fig 4.

3.2.3. Inception-ResNetV2: Inception-ResNetV2[14] takes the input size similar to that of Inception-V3 model, that is 299 x 299 RGB images. Each Inception block is followed by filter expansion layer which is used for scaling up the dimensionality of the filter bank before the
addition to match the depth of the input. Inception-ResNetV2 uses batch-normalization only on the top of the traditional layers and not on the top of the summation. Different levels of Inception-ResNetV2 model is shown in Fig 5.

3.3. Decoder: LSTM
To get the actual comparison between the feature extracting models, common decoder is used for all the above mentioned encoders, that is single layer LSTM followed by hidden neural layers. LSTMs are generally used to remember the information of the previous output. The key idea of using LSTM is how it controls the information. The cell, in LSTM, acts as the transport highway, so the information from the earlier time step can make its way to the next time steps. The state of the information, whether it gets added or removed, depends on the gates of the cell. The gates are composed of sigmoid functions (it varies from 0 and 1), so the cell can decide how much information it has to pass to the next state of the layer, for 0, it means the information is removed, and for 1, the whole information is passed onto the next cell.
Fig 6 shows how the information flows in a LSTM cell.

Gates can be divided into 3 types: forget gate, input gate and output gate. Forget gate decides the state of the information, whether it should be kept or deleted. Previous hidden state and current state is passed into the input gate to update the state of the cell. State of the next cell is decided by the output gate.

Equations of LSTM are written below:

\[ i = \sigma(x_t U_i + s_{t-1} W_i) \]  
\[ f = \sigma(x_t U_f + s_{t-1} W_f) \]  
\[ o = \sigma(x_t U_o + s_{t-1} W_o) \]  
\[ s_t = \text{tanh}(c_t) o \]
Parameters of equations (1) to (5) are as defined:
1- i is the input gate.
2- f is the forget gate.
3- o is the output layer.
4- W is the connection of the previous.
5- U connects the input to current hidden layer.

4. Training
4.1. Experiment
4.1.1. Evaluation Metric BLEU score is used for evaluating the quality of caption generated by the algorithm based on the actual captions. Scores are calculated for individual translated segments by comparing them with a set of good quality reference translations.

4.2. Method
Due to the equipment limitations, only one caption is utilized, out of five captions. The composition of dataset is kept same as was previously defined, 6k for training, 1k for validation and 1k for testing.

In the encoding models, which is convolutional models in this paper, the output is restricted upto the second last layer of the model. Training parameters are listed in Table 1.

| Table 1. Training Parameter |
|-----------------------------|
| Learning Rate | 0.08 |
| Optimizer      | RMSprop |
| Batch Size     | 512 |

5. Result
Structures of different encoding models are discussed in section III. Features from images are extracted in the form of vectors using these models. The extracted features are then used in the decoding layer of LSTMs which generated the captions of images.

Performance of encoding models are different on the same images. Therefore, we used the best and worst BLEU scores produced by the models, that might be on different images, to evaluate the results.

5.1. VGG16
VGG16 is most basic of all the encoders discussed. It uses a basic 3X3 filter size for all the convolutional layers.

Figure 7. VGG16 best BLEU score: 0.84053816
The best BLUE score is least for VGG16 since it lacks the depth as compared to the VGG19 or the variety of filters as compared to Inception-V3 and Inception-ResNetV2. The best and worst BLEU scores for VGG16 layer are shown in Fig 7 and Fig 8, respectively.

5.2. VGG19
As VGG19 has more depth in convolutional layers in comparison to the VGG16 (which resulted in the much better feature extraction of image), that resulted in the better BLEU score for the VGG19 model. The best and worst BLEU scores for VGG19 layer are shown in Fig 9 and Fig 10, respectively.

5.3. Inception-V3 and Inception-ResNetV2
The range of BLEU score is least in case of Inception-V3 because of the different filters available in the model that can extract more features as compared to the VGG16 and VGG19. The best BLEU score for Inception-V3 and Inception-ResNetV2 are shown in Fig 11 and Fig 12, respectively. Image with the worst BLEU score for both Inception-V3 and Inception-ResNetV2 is shown in Fig 13.

In the results, it was observed that the models did not perform better for some complicated images. Either there is too many different information (pixel values) in the image, or the information are too similar, this made the task of encoding models difficult to extract features from the images.
Figure 10. VGG19 worst BLEU score: 2.77554679e-78

Figure 11. Inception-V3 best BLEU score: 0.895487043

Figure 12. Inception-ResNetV2 best BLEU score: 0.882496

6. Conclusion
In this paper, with the help of keras and tensorflow, Flickr8k dataset is trained using different encoding models to obtain the desired result. Inception-V3 is the better model amongst all the other models discussed in section III, as the range of the BLEU score is least for Inception-V3. These are all basic models that can be used for most of the image captioning task, as most of the captioning task involves the feature extraction from image and using these features for text generation.
7. References

[1] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Cyrus Rashtctian, Julia Hockenmaier and David Forsyth. Every Picture tells a story: Generating sentences from images. In Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV’10, pages 15-29, 2010.

[2] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan 2015 Show and Tell: A Neural Image Caption Generator arXiv:1411.4555

[3] Fang Fang, Hanli Wang, Pengjie Tang 2018 Image Captioning with Word Level Attention 25th IEEE International Conference on Image Processing (ICIP), pages 1278-1282

[4] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, Svetlana Lazebnik ICCV, 2015 Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models.

[5] https://forms.illinois.edu/sec/1713398

[6] http://cocodataset.org/download

[7] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell 2014 Long-term recurrent convolutional networks for visual recognition and description. CoRR, abs/1411.4952.

[8] Andrej Karpathy and Fei-Fei Li 2014 Deep visual-semantic alignments for generating image descriptions. CoRR, abs/1412.2306.

[9] Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dolla’r, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig 2014 From captions to visual concepts and back. CoRR, abs/1411.4952.

[10] Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra 2015 DRAW: A recurrent neural network for image generation. CoRR, abs/1502.04623.

[11] Parth Shah, Vishvajit Bakrola, Supriya Pati 2017 Image captioning using deep neural architectures International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pages 1-4

[12] Karen Simonyan and Andrew Zisserman 2014 Very deep convolutional Networks for large-scale image recognition arXiv:1409.1556.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna 2016 Rethinking the Architecture for Computer Vision IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818-2826

[14] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi 2016 Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning arXiv:1602.07261.