Study of types of Image Captioning Techniques

Abhishek Singh
(202118004)
DAIICT
Msc DS
202118004@daiict.ac.in

Rachit kumar singh
(202118015)
DAIICT
Msc DS
202118015@daiict.ac.in

Devarsh Antani
(202118044)
DAIICT
Msc DS
202118044@daiict.ac.in

***Abstract*—Image captioning refers to generating captions automatically for a given image. As a recently emerged research area, it is attracting more and more attention. To achieve the goal of image captioning, semantic information of images needs to be captured and expressed in natural languages. Connecting both research communities of computer vision and natural language processing.**

1. Introduction

Image caption generation is a problem in which a model is trained to generate a natural language description of an image. This is a challenging task that involves both understanding the content of the image and generating coherent and fluent language to describe it. Deep learning models have been shown to be effective for image caption generation. These models typically use a convolutional neural network (CNN) to extract features from the image, and a recurrent neural network (RNN) to generate the natural language description. One such model is the encoder-decoder architecture, in which the CNN acts as an encoder to extract features from the image and the RNN acts as a decoder to generate the caption. The model is trained on a large dataset of images and their corresponding captions, using techniques such as supervised learning and reinforcement learning. Another popular approach for image caption generation is the use of attention mechanisms, which allow the model to focus on specific regions of the image when generating the corresponding words in the caption. This can improve the accuracy and fluency of the generated captions. Overall, deep learning models have shown promise for the task of image caption generation, and continue to be an active area of research.

In our project we have demonstrated image captioning using 3 different deep learning models 1) Image captioning using multi-model simple RNN 2) Image captioning using attention mechanism 3) Image captioning using ResNet and LSTM, although all of them demonstrated using the same dataset.

2. Dataset

We've used Flickr8k dataset. It was created by researchers at the University of Maryland, College Park and consists of 8,000 images taken from the website named Kaggle [4], each paired with five different captions that describe the scene or subject portrayed in the image.

3. Image captioning with multimodal simple RNN:

To generate novel captions with multimodal simple RNN we've implemented a research paper from, Mao et al.[1]

A simple recurrent neural network (RNN) is a type of recurrent neural network that uses a single layer of neurons and weights to process input sequences and generate output sequences. This type of network is called "simple" because it has a relatively straightforward architecture compared to other types of RNNs, which can have multiple layers of neurons and more complex connections between neurons.

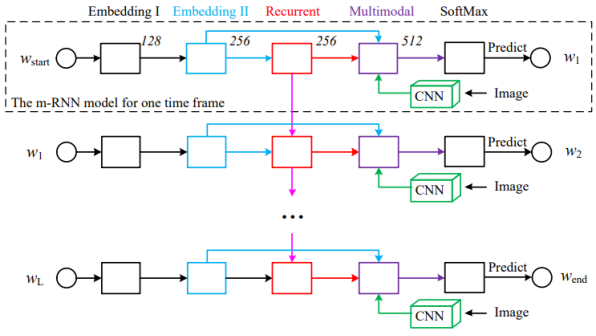

Figure 1. Multimodel RNN

3.1. Data Preprocessing

1. Tokenization: Splitting the text into individual words or sequences of words (tokens).

2. Stop word removal: Removing common words that add little meaning to the text, such as ”a,” ”an,” and ”the”, etc.

3. Lower Case: Converting all words to lowercase to reduce the number of unique words in the dataset.

4. Token added padding: Adding special tokens to the beginning and end of each sequence to indicate the start and end of a caption.

After that we took out the list of words and its length which is 8683 which will be used in model training further and also maximum length of caption 36 which will be used further for padding.

3.2. Feature Extraction

For this problem statement we've used VGG16 and AlexNet models to extract features from images. We need this feature extraction step because it is a part of the dimensionality reduction process. VGG16 is a part of VGG models which stands for Visual Geometry Group, it is a standard deep Convolutional Neural Network (CNN) architecture which supports 16 layers of VGG. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers. AlexNet is another deep learning architecture which consists 8 layers of learnable parameters, which consists of 5 convolutional layers,3 max pooling followed by 3 fully connected layers.

Both architectures have used 224*224*3 dimension for input image and gave the output of 4096 dimension which will be later used in our fully connected layer. Although VGG16 generates about **134,260,544** parameters while AlexNet generates about **88,661,888** parameters and extracted about 8091 features. All the extracted features are saved in pickle files which can be used later so we don't have to extract features every time which is very time consuming.

3.3. Model Training

Before the model training, we've splitted the extracted features into 90% training and 10% for test samples which is the 7281 of training and 810 of testing samples out of 8091 features.And as mentioned above we got vocabulary of 8683 and max length of caption was 33.

Generator function is part of model training which will take data in batches of size 32 for a single image rather than the entire dataset. The function has input sequence for the caption and it will be padded by maximum length of our vocabulary while output sequence does one hot encoding. All the image extracted features will be appended to lists, those would be used by model training later.

There is a model training function defined which has encoding layers and decoding layers, for encoding layers it has 2 types one for image and other one for caption. The Encoder layer for image has input shape of 4096 dimensional shape, using dense function with 256 units with Relu activation function and in encoder text layer has input shape of vocabulary maximum length which is 33, embedding layer which enables us to convert each word into a fixed length vector of defined size which is our case it is 8683, simple RNN function which takes 256 unites. To reduce overfitting we've also used drop out layer.The decoder layer which takes combination of both image encoder and text encoder and it has one dense layer with 256 units with Relu activation function and output layer size of vocabulary length(8683) with softmax activation function.

Model compilation is a part of training where we provide loss function, optimizer and other hyperparameters. Here loss function is categorical cross entropy with adam optimizer, epoch is 20, batch size 32. for each epoch we called generator function which is defined above.

3.4. Evaluating with multimodel simple RNN

In our project we've used the BLEU (bilingual evaluation understudy) score to evaluate the quality of generated text. The BLEU score can be used to evaluate the quality of the captions generated by a model.

For evaluation in the table below you can see VGG16 is much better than AlexNet because it is more accurate and has better performance on a wide range of image recognition tasks. This is because VGG16 uses a deeper network architecture than AlexNet, with 16 compared to AlexNet's 8 layers. This increased depth allows VGG16 to learn more complex and abstract features from the data, leading to better performance. Additionally, VGG16 uses smaller convolutional filters, which allows it to capture more information from the images.

	VGG16	AlexNet
BLEU1	0.54	0.48
BLEU2	0.31	0.19
BLEU3	0.22	0.11
BLEU4	0.11	0.04

Predicted:

startseq black dog with red collar running in field endseq

Figure 2. predicted sample image

3.5. Visual Result:

4. Image Captioning using ResNet and LSTM

This section is inspired by the research paper [2].

4.1. Introduction

The encoder–decoder framework is adopted to generate captions for images. The framework is originally designed to translate sentences from one language into another language. Motivated by the neural machine translation idea, it is argued that image captioning can be formulated as a translation problem, where the input is an image, while the output is a sentence. In image captioning methods under this framework, an encoder neural network first encodes an image into an intermediate representation, then a decoder recurrent neural network takes the intermediate representation as input and generate a sentence word by word.

4.2. Data

4.2.1. Data Cleaning. The text data does not require any cleaning because removing the punctuation or numbers or converting the text to lowercase will develop large errors and result in a less accurate model. Same is applicable for the images.

4.2.2. Data Preprocessing. First step is to split the sentences into words. To convert words to vectors, we've used a method called 'one hot encoding', which converts the words to its corresponding vectors based on the index of the word in that particular corpus.

4.3. MODEL

4.3.1. Feature Extraction. For extracting the features from the images, we used a CNN architecture called ResNet 50.

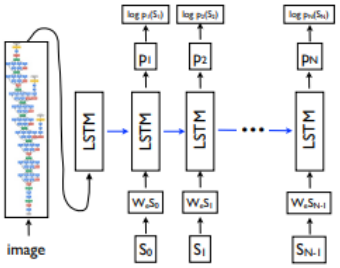

Figure 3. Model Architecture

ResNet 50 is a pre-trained transfer learning architecture with 50 layers with convolution layer, pooling layers, combined. But since we only require to extract the features and now to predict any output, we remove the last predicting layer from the model architecture.

So the process begins with a raw image, the image can be converted into an RGB format(if not already, for better results). Then, the resizing of the image is required since the input size for an image required to be processed in ResNet is 224 x 224.

After applying the following steps, the transformed image is given to the ResNet 50 model and we get the required set of vectors as output from the model.

4.3.2. Data Preparation. After converting the words to vectors (which has already been discussed) we find out the maximum length in a particular sentence. The maximum number of words in a particular caption was 36. Then 'post' padding has been applied ('pre' padding can also be used) to make the length of all the sentences to the same length.

4.3.3. Training Data. Due to limited resources, we've used 1500 images corresponding to 7500 captions.

4.3.4. Model Building. Input format
As of now, we have:
1. Extracted Image features.
2. Transformed captions in vector format.

The Extracted Image features are then passed through a dense layer with the shape same as the input shape and activation function as 'ReLu' (since, works well in hidden layers).

The Transformed captions are passed through a single LSTM layer with 256 neurons. The input format for the model is such that we take the output of the above .So we concatenate these two to make a single unit.

Final Model:

The model building consists of the final 2 layers of LSTM. First LSTM layer with 128 neurons (128 because mostly preferred). The second LSTM layer has 512 neurons

(512, because a more complex model was required since the model was under fitting) with batch size of 512 and 100 epochs. We saw that our model could have achieved a better result if we had experimented more on the batch size and number of epochs but due to computational limits we couldn't experiment much.

Finally, we use 'Softmax' in our last layer (softmax since it is a many to many rnn problem statement hence more than 2 outputs are given).

The choice of loss function has been 'Categorical Cross Entropy' (because we have categorical output) and the optimizer is 'RMSprop' which resulted in achieving a better accuracy for our problem statement.

4.3.5. Model Result.

	Flcikr8k	Flcikr30k
BLEU1	0.31	0.35
BLEU2	0.21	0.22
BLEU3	0.13	0.10
BLEU4	0.12	0.08

4.4. Visual Result

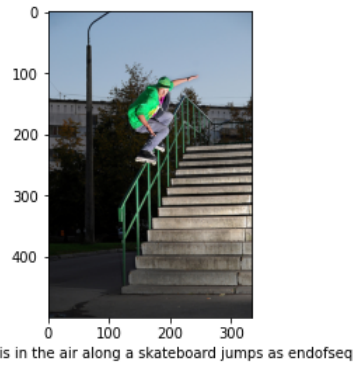

Figure 4. Image depicting results from the trained model

5. Model training : Neural Image Caption Generation with Visual Attention

In this section, we've implemented a research paper from, Xu et al.[3]

In the previous works of image captioning models, it did not use the idea of attention that is essential in how we generate captions from images. As we know that image contains a lot of information but human visual perception systems give attention to the salient features in the image. So using attention, we narrow down all the information of the image into its salient features to understand the image. This is especially needed when there is a lot of information in the image which is not important. So We need a model that uses attention to generate captions that contains only salient features in the image but in doing so not lose information that can be helpful in generating more rich and descriptive caption. In this method, we use deep learning techniques like CNN and LSTM with visual attention to generate captions. We use an attentive encoder - decoder model where the encoder is the CNN and the decoder is the LSTM with an additional attention layer between encoder and decoder.

5.1. Data preprocessing

We used flicker8k dataset which contain 8091 images each with five captions therefore total 40455 captions. Instead of using all 40455 captions to train the dataset, we took 40000 captions with its image. First, we converted all the words in the captions into lowercase, then we removed all the punctuation and numerical value from the caption. We also added 'start' at the start of every caption and 'end' at the end of every caption. We use library "tokenizer.textstosequences" to convert the words into numerical values that represent the index of the word in the caption.

5.2. Encoder: features extraction using CNN

In this method, first we pass the image through CNN to generate features of the image. The image has dimension of 224*224*3. To do so we use transfer learning where we use pre trained VGG16 where we do not use the final softmax layer. This generates feature vectors which we call annotation vectors. This produces N vectors, each of which is a D-dimensional representation of part of the image.

5.3. Decoder: caption generation using LSTM

The LSTM layer is used to generate words of the captions at every time step which depends on the context vector which is calculated in the attention layer, previous hidden state and previously generated words. As LSTM is capable of learning long term dependency and as the word generated at each time step is dependent on previous generated words therefore LSTM is used as a decoder to generate captions.

The LSTM used in this method is given in the figure 5 LSTM Cell.

and formulas used in this LSTM are given in figure 2 LSTM Formula.

5.4. Training

the neural network model was trained using adam optimizer and the loss function used was SparseCategorical-Crossentropy that was used in backpropogation. 512 units of LSTM was used in training and activation function used in the LSTM neural network was tanh. the batch size was 64 and epoch was 20.

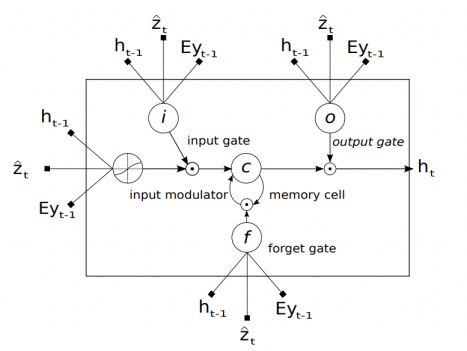

Figure 5. LSTM Cell.

$$i_l = \sigma(W_{ix}x_l + W_{im}m_{l-1} + W_{iz}z_l),$$

$$f_l = \sigma(W_{fx}x_l + W_{fm}m_{l-1} + W_{fz}z_l),$$

$$o_l = \sigma(W_{ox}x_l + W_{om}m_{l-1} + W_{oz}z_l),$$

$$c_l = f_l \odot c_{l-1} + i_l \odot h(W_{cx}x_l + W_{cm}m_{l-1} + W_{cz}z_l),$$

$$m_l = o_l \odot c_l.$$

Figure 6. LSTM Formula.

5.5. Attention layer

We used an attention layer to calculate context vector zl which is function of image region vectors (a1 , . . . aN) and weights associated with them. $(\alpha_1, ...\alpha_N)$:
$z_l = \phi(a_i, \alpha_i)$

By doing this, we are able to give attention to different parts of the image differently depending on the weights calculated. There are two attention mechanisms depending on function forms used. these attention mechanism are

5.5.1. Soft Attention. In the soft attention mechanism, the positive weights $a_{l,i}$ associated with location i at time step l is used to represent the importance of location i of the image vector in calculating the context vector. To calculate these weights at each time step t, we used a multilayer perceptron conditioned on the previous hidden state h_{t1}. The hidden state changes as the LSTM moves forward to calculate output sequence. So the model can be represent in formula as

Once weights are computed, we can calculate the context vector as it is weighted sum of each image vector,therefore the formula is

$$z_l = \sum_i^N \alpha_{l,i} a_i$$

5.5.2. Hard Attention. At each time step in hard attention mechanism, we select only one of the N vectors of the image vector to generate word.At time step t, for each location i, the positive weight $\alpha_{t,i}$ associated with it is taken as the probability for this location to be focused on for generating the corresponding word. The context vector z_t is calculated as follows:

$$z_t = \sum_i^N s_{t,i} a_i$$

where $s_{t,i}$ is an indicator variable, which is 1, if the visual feature a_i from the i_{th} location out of N is attended at time step t, otherwise 0. The distribution of the variable $s_{t,i}$ is treated as a multinoulli distribution parameterized by a_i, i and its value is calculated based on sampling.

5.5.3. Evaluation. I was able to run my model on 2 image that I did not use in training the model to calculate the BLEU score which is used as measure of performance of the model.

In blue score, the real caption and the predicted caption of the 2 test image are shown in the image below

```
BELU score: 54.10822690539396
Real Caption: Two white dogs are playing in the snow
Prediction Caption: two white dogs are playing in snow
```

Figure 7. performance measure.

```
BELU score: 35.49481056010053
Real Caption: girl doing back bend
Prediction Caption: little girl is doing back bend in field
```

Figure 8. performance measure.

6. Conclusion

In conclusion, image caption generation is a task in which a model is trained to generate a natural language description of an image. This task has many potential applications, including assisting visually impaired individuals, improving image search engines, and providing additional context for images in social media and other online platforms. There are many different approaches to image caption generation, including using convolutional neural networks (CNNs) and recurrent neural networks (RNNs), either alone or in combination. The quality of generated captions can be evaluated using metrics such as the BLEU score. Overall, image caption generation is an active area of research with many exciting developments and potential applications.

References

[1] J. Mao, W. Xu, Y. Yang, J. Wang, A.L. Yuille, Explain images with multimodal recurrent neural networks, arXiv:1410.1090v1 (2014).

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

[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, arXiv:1502.03044v3 (2016)

[4] Flickr 8k Dataset