Image Captioning on Pie Charts

Kunlanit Korsamphan
Information Management
Asian Institute of Technology
Pathum Thani, Thailand
st121422@ait.asia

Shin Thant
Data Science and AI Program
Asian Institute of Technology
Pathum Thani, Thailand
st121493@ait.asia

Phway Thant Thant Soe Lin
Data Science and AI Program
Asian Institute of Technology
Pathum Thani, Thailand
st121494@ait.asia

Generating a description of an image is called image captioning. Image captioning requires to recognize the important objects, their attributes and their relationships in an image and generate syntactically and semantically correct sentences. In this paper, manually generated pie chart images are used to generate the descriptions of these pie charts automatically. Pie charts are important representations of data collections. Deep learning techniques: CNN and RNN are used for encoding and decoding. Here, we also use the Bilingual Evaluation Understudy also known as BLEU for accuracy calculation of the text generated. The average BLEU score is 0.42 which can be said as the result captions are high quality captions. Based on the result, our model may need to improve by adding more training data. At a certain point, our model will be a very useful tool for financial related works.

***Index Terms*—CNN, image captioning, LSTM, pie chart captioning, RNN**

I. INTRODUCTION

IMAGE captioning is a task where machines learns to generate a caption or a description for an image. It displays the ability of computers that can understand the image and its relationship with a natural language such as English. This task has actively been performed on images of people, activities, animals, nature, etc.

Pie chart is a simple and widely used chart in many fields. It visualize how different sectors make up a whole. Describing a pie chart can be done by comparing between each sector to determine what share of the total each sector has. Data interpretation for it is important since people need to get the right information for each category. Some pie charts are hard to interpret because of the lack of information like it does not provide the percentage properly. If machines can interpret pie charts and give the appropriate description for people, these interpretation problems can be overcome and it will be faster in working areas that need to deal with a lot of data representation stuff.

In our project, we specifically do image captioning on pie charts by trying to experiment on the existing models. Our main objectives is to understand the performance of models on images of pie chart where it consists of the chart, the legend and some texts as labels. Next, we try to explore the effect of different CNN architecture and the different structure on input captions have on the output caption. We try to implement the captioning model with 2 different framework of python which are Keras and PyTorch. This implementations have different model architecture and parameters.

The data sets for this project is manually created by ourselves as there is insufficient data sources for training the model.The task was performed with ResNet50 and VGG-16 model as an encoder to vectorized and extracted image features and LSTM model as a decoder to generate a caption.

II. RELATED WORK

Many applications have motivated the study of data extraction from scientific charts in various contexts. The extracted data are described in captions. Most of the research works on charts captioning is done through the combination of machine learning approach and image processing techniques.

A machine learning based system extracts and recognizes the various data fields present in a bar chart for semantic labeling [1]. The approach comprises a graphics and text separation and extraction phase, followed by a component role classification for both text and graphic components that are in turn used for semantic analysis and representation of the chart [1].

The Chart-Text fully automated system creates textual description of chart images [2]. Given a PNG image of a chart, the Chart-Text system creates a complete textual description of it [2]. First, the system classifies the type of chart and then it detects and classifies the labels and texts in the charts using transfer learning [2]. They use MobileNet as a classifier and Faster R-CNN as a detector [2]. Finally, it uses specific image processing algorithms to extract relevant information from the chart images [2]. They randomly generated their data set using Matplotlib [2].

The prior researches are not pure machine learning solutions for chart captioning. There are also researches that use purely machine learning techniques for image captioning.

Convolutional Neural Network(CNN) and Recurrent Neural Network (RNN) is the most common framework among them. To increase efficiency, much of the work aims to change the language model and the relation between the vision and language models. In this network, the deep convolutional network is used to extracted features of images and sentence generation is done by a powerful RNN which is trained with the visual input so that RNN can keep track of the objects explained by the text. The m-RNN model uses a vanilla RNN combined with different CNNs [3]. The RNN hidden state and the CNN output are fed into a multimodal block at each time point to integrate the image and language features, and the next term is predicted by a softmax layer.

For a neutral image captioning (NIC) model, it uses LSTM as a decoder. The NIC model takes the image function

vector as an input at the outset, and then the visual data is transferred along the recurrent path [3]. In both m-RNN and NIC, one image is represented with a single vector and ignores the different areas and objects in the image. Later, a spatial attention mechanism is introduced to pay attention to different areas at each time step in image captioning model [3]. Hierarchical LSTMs have also been used to generate a paragraph description consisting of multiple sentences to describe an image [3].

In CNN + CNN model, there are four modules: (1) vision module is adopted to observe the images;(2) language module is to model sentences; (3) attention module is for connecting the vision module with the language module; (4) prediction module takes input the visual features from the attention module and concepts from the language module and predicts the next word. The vision CNN extracts features from the image, and the language CNN is applied to create the sentence [4]. Meanwhile, the attention module and prediction module fuse the information from image and word contexts [4]. To allow feed-forward sentence generation,the convolutions in the CNN have been used as causal filters. A CAN without pooling is the basis of the language model, which is very distinct from the traditional RNN-based system. To Memorize the context, RNNs follow a recurrent path, while CNN's use kernels and stack several layers to model the context [4]. For this experiment, they used the Flickr30k and COCO data set [4].

Image captioning can also be performed with an attention based model that automatically learns to describe the content of images [5]. They used a Convolutional Neural Network in order to extract a set of feature vectors which are referred to as annotation vectors [5]. They used a long short-term memory (LSTM) network that produces a caption by generating one word at every time step conditioned on a context vector, the previous hidden state and the previously generated words [5]. They used both two alternative mechanisms for the attention model stochastic 'Hard' attention and deterministic 'soft' attention [5].

III. Methodology

A. Data Preparation

As many data sources for image captioning do not have data on pie charts, we needed to manually generated the image of pie charts by ourselves. The total number of data is 1290 images with caption for each image. The target caption for each image was generated based on the patterns that mostly used to describe pie chart. We tried to generate 2 type of captions where one consists of numerical data that related to the chart (original) and the other one is a caption that has less numerical data but using the several quantitative words to describe pie chart sector (modify).

B. Data Cleanup and Pre-process

Due to the randomness of caption, it need to be cleanup and pre-process before using it as an input to the captioning model. First, all caption will need to be change to lowercase. The punctuation will be removed from the sentence except the percentage sign. The best practice for normal image captioning case is remove all numerical data from the caption, but, as our goal is describing a chart, it is unavoidable for number to exist in the caption.

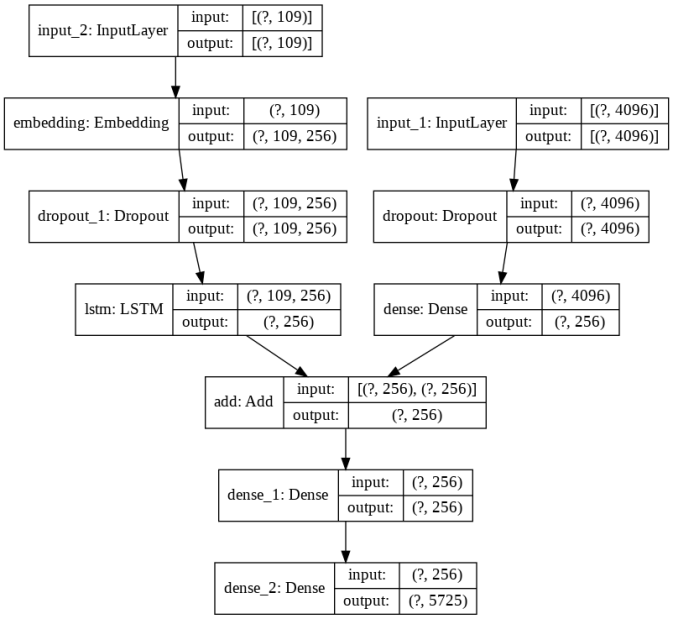

Fig. 1. VGG-16 Model with LSTM on Keras framework

After clean up the data, we need to create vocabulary or dictionary for our caption. The dictionary is a set of words that are used to describe the data set caption. The dictionary consists of 2 parts which are a data of key-value pair of all words where a word is a key and a number is a value represent the index of that word in dictionary, and another key-value pair where the index is a key and the word is a value. We also need to add <start> and <end> as tags to indicate the start and end of the caption when the model generate it, <null> as a padding tag, and <unk> as an unknown tag for out of index value.

Dictionary is used to vectorized caption after we tokenized it. Tokenized caption is a process that break the sentence into a list of word by its order. After tokenized the caption, it needed to be vectorized by mapping each word in the list with its index in dictionary. The whole list needed to be padded to have an equal length by adding <null> tag to the tokenized list and also convert it to its index in the dictionary.

The vectorized caption will be an input to our model for caption prediction. Another input for our model is image features. In this project we used 2 CNN architecture to help us extracting the features; ResNet50 and VGG-16. Normally these 2 architectures are used on classification tasks. To use it in our task, we need to removed the last 2 layers from current architecture. The final shape of ResNet50 is (batch number, 2048) and the final shape of VGG-16 is (batch number, 4096)

C. Model

Since we use 2 different frameworks to implement the captioning model, the process to extract image features was

X1,	X2 (text sequence),	y (word)
photo	startseq,	little
photo	startseq, little,	girl
photo	startseq, little, girl,	running
photo	startseq, little, girl, running,	in
photo	startseq, little, girl, running, in,	field
photo	startseq, little, girl, running, in, field,	endseq

Fig. 2. Tokenized captions and its sequences for Keras training input.

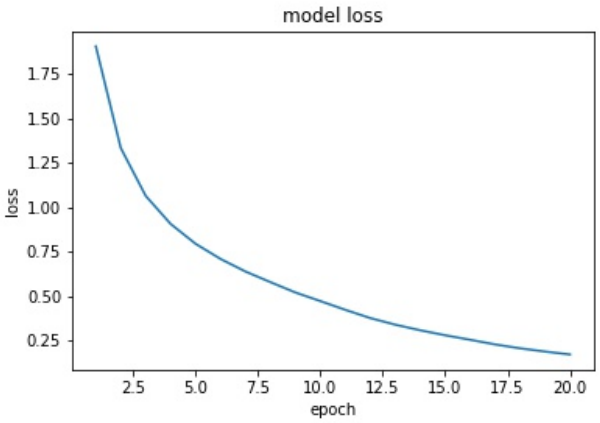

Fig. 3. PyTorch: Loss value of VGG-16 on full caption with numerical value .

also varied. The figure 1 shows the structure of the VGG-16 as feature extractor model on Keras where the input image will be processed and have the output as (batch numbers, 256). Another input is caption or sequence data with length of 109 words, will be embedding first then processed through LSTM to have a final shape as same as the image features. The dropout layer were added to help with over fitting model. These 2 inputs then will be merged and have a final shape as the total words in dictionary. The layers for ResNet50 on Keras is the same with its processes. The only difference is the input shape of the encoder, which is (batch number, 2048).

For PyTorch implementation, the encoder are quite the same with additional embedding layer on image to have an output shape as (batch number, 300) for both CNN architecture. The decoder takes image and vectorized captions as an input. It merged the 2 inputs together ans go through LSTM layer to decode it, which is different from the Keras framework where only caption will go through LSTM.

D. Training

To train the model, the data set was split into 70% for training and 30% for testing. It will run with epoch value equals to 20 for all model. PyTorch framework implementation can accept image feature and its vectorized caption directly. It also implement progressive loading to help with memory error during the execution.

On the other hand, Keras needs to generate word sequences as shown in figure 2. The model will take image feature, the sequence of vectorized caption and the next target word as training input.

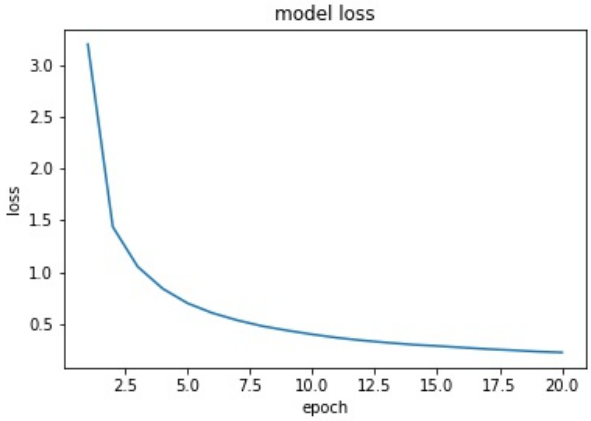

Fig. 4. PyTorch: Loss value of VGG-16 on less numerical caption .

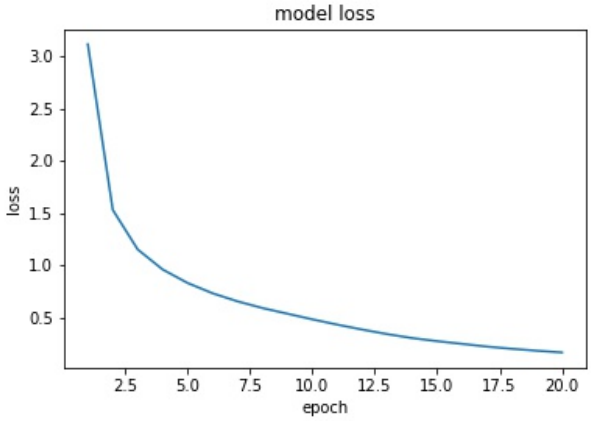

Fig. 5. PyTorch: Loss value of ResNet50 on full caption with numerical value .

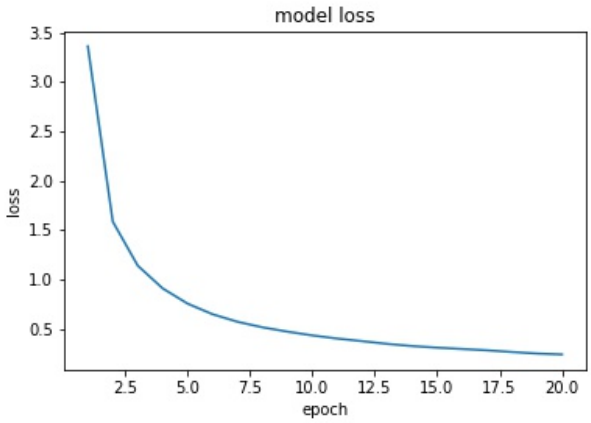

Fig. 6. PyTorch: Loss value of ResNet50 on less numerical caption .

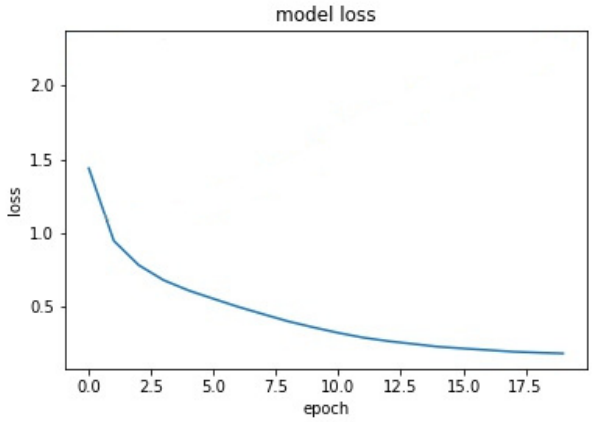

Fig. 7. Keras: Loss value of VGG-16 on full caption with numerical value .

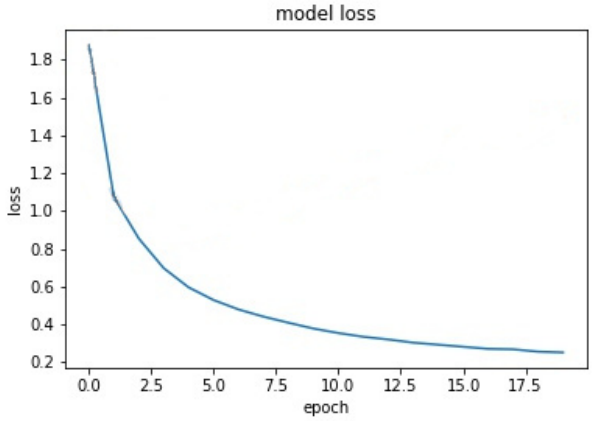

Fig. 8. Keras: Loss value of VGG-16 on less numerical caption .

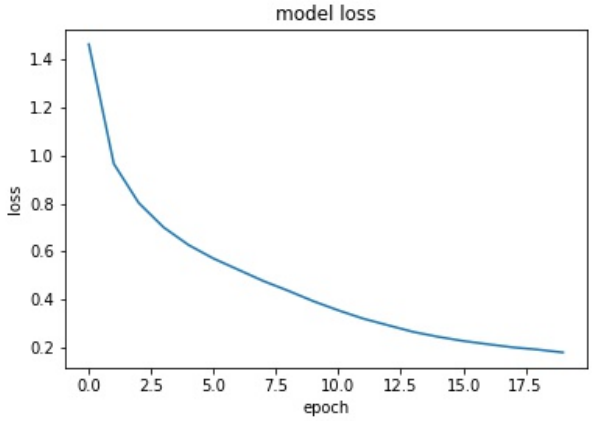

Fig. 9. Keras: Loss value of ResNet50 on full caption with numerical value .

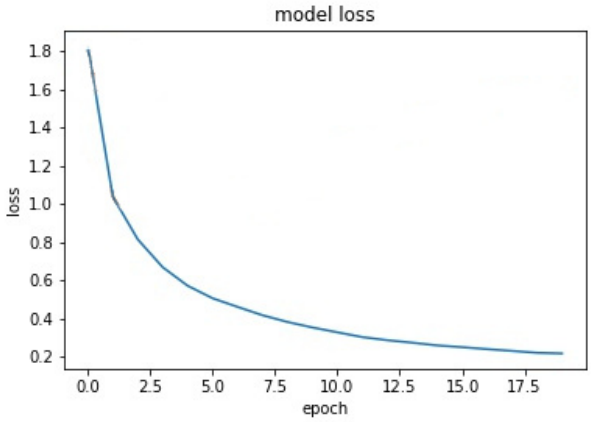

Fig. 10. Keras: Loss value of ResNet50 on less numerical caption .

From figure 3 to figure 10, we could see that the loss value for all models are quite similar even though there are differences in architecture and CNN model.

E. Testing

Testing the model occurs after training by using 30% of data set. The starting input for testing is image feature and the ¡start¿ tag as a caption. The logic behind caption prediction is the greedy search. The model will perform a calculation and find the most likely to be the next word by from maximum value. When the next word was predicted, it will be append to the sequence and used as an input to predicted the next word until it reach the <end> tag or reach maximum length.

IV. Evaluation

Once the model is trained, we can evaluate the model by generated captions from train data set and evaluated the predictions by the BLEU score. The BLEU score are used in text translation for evaluating translated text against one or more reference translations. As long as the machine is generated pretty close to any of the references provided by human, the BLEU score will be high[6].

$$BP = \begin{cases} 1 & \text{if } c > \text{r} \\ e^{1-r/c} & \text{if } c \leq \text{r} \end{cases} \tag{1}$$

$$BLEU = BP \times \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{2}$$

The actual and predicted descriptions are evaluated by sentence BLEU score. The figure 11 display the summary of the BLEU score on different models and different caption structure. From the table, we could see that a model implement on PyTorch with ResNet50 as CNN and the input caption included numerical data get the highest average score. The model implement on PyTorch with VGG-16 as CNN and the input caption has less numerical data get the lowest average BLEU score.

	Avg BLEU	Min BLEU	Max BLEU
Pytorch_ResNet50_original	0.4615335	0.127573798	0.6946034
Keras_VGG16_original	0.4279141	0.1855183	0.74519
Pytorch_VGG16_original	0.4225741	0.1453339	0.6614093
Keras_VGG16_Modify	0.4192764	0.1228039	0.8288602
Keras_ResNet50_Modify	0.411443	0.1450969	0.8086343
Pytorch_ResNet50_Modify	0.4038637	0.1922471	0.685348
Pytorch_VGG16_Modify	0.3850448	0.1741087	0.675286
Keras_ResNet50_original	0.3811387	0.1463759	0.745711

Fig. 11. The result of each model and the summary on BLEU score.

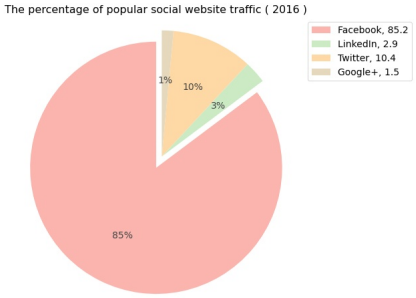

Fig. 12. The predicted caption with low BLEU score.
Actual: this chart shows the percentage of popular social website traffic in 2016 the highest percentage of traffic is facebook 85.22 while linkedin has a the lowest percentage at 2.86
Predicted: this graph illustrates the number of college in italy by level the maximum value is 2-year which made up 228.0
BLEU Score: 0.145333870342886.

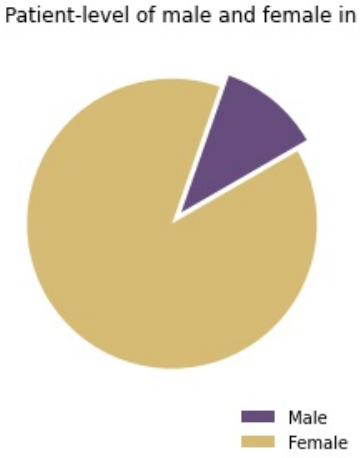

Fig. 13. The predicted caption with high BLEU score
Actual: this pie chart illustrates patient-level of male and female in female is by far the largest proportion which larger than 70%.
Predicted: this chart illustrates patient-level of male and female in male is by far the largest proportion which larger than 70%.
BLEU Score: 0.828860205.

V. Discussion and Conclusion

We have implemented a caption generator for pie charts. We have tried generating the pie charts manually by using Matplotlib library of Python. It gives a description of the pie charts in English language. With the use of convolution neural network which is used as an encoder for image input and later a recurrent neural network which helps in generating text for the image. We have used ResNet50 and VGG-16 as our encoder and for the decoder we have chosen LSTM. We have collected a total of 1290 pie charts which we have divided into 903 training images and 387 testing images. The use of BLEU score is implemented to check the accuracy. The average BLEU score we got is 0.4615. For future work, we could add more intricate step by step images of different pie chart sectors. In current situation, the captions we use as training are a little bit complex and long. We can improve the accuracy by preparing more flexible ground truth captions. We believe this will improve the BLEU score as most of the generated captions are correct.

Since BLEU is n-gram based, the score will depend on similarity with ground truth caption. So, score might be lower although the predicted caption is correct for the given image. The more epochs we use for training, the more chance that the model can be over-fitted. If we train the model with 500 epoch, around 300 epochs, the model becomes overfit with 0.9 BLEU score and it gives same result with trained captions on respective trained images. (loss is around 0.0004 at that stage). Our pie chart captioning model will be a useful application for office stuff or any other procedures in which charts play a crucial part.

References

[1] Rabah A. Al-Zaidy, C. Lee Giles, " A Machine Learning Approach for Semantic Structuring of Scientific Charts in Scholarly Documents", Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Feb 2017. pp. 4644–4649.

[2] Abhijit Balaji, Thuvaarakkesh Ramanathan and Venkateshwarlu Sonathi, "Chart-Text: A Fully Automated Chart Image Descriptor", Computer Vision and Pattern Recognition, Dec 2018. Available doi: arXiv:1812.10636.

[3] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)", ICLR,2015.

[4] Q. B. Wang and A. B. Chan, "CNN Hengelo: Convolutional Decoders for Image Captioning," Computer Vision and Pattern Recognition, May 2018. Available doi: arXiv:1805.09019.

[5] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel and Yoshua Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", Machine Learning, Apr 2016. Available doi: arXiv:1502.03044.

[6] K. Papineni, S. Roukos, T. Ward, and W. Zhu,"BLEU: A Method for Automatic Evaluation of Machine Translation," In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL), Philadelphia, 2002, pp. 311-318.