Generating Chinese Captions for Flickr30K images  
Hao Peng, Nianheng Li

Introduction  
Recently, many researches on image caption tasks have been devoted to RNN models. However, all these models are trained on images with English captions. Thus we don’t know their performance in other languages to see that whether this method works universally.

In this work, we trained a RNN on Flickr30K dataset with Chinese sentences. As Chinese sentence has no space between words, we implemented the model with the same architecture used by [2] on Flickr30 in two methods. In the first setting, we used Jieba[5] package to tokenize each Chinese sentence into a list of words and feed them to the RNN. While in the second one, we split each Chinese sentence into a list of characters and feed them into the same model. We report the BLEU score of our two methods and also compare them to the result achieved by [2].

Data & Processing  
We trained our model on Flickr30K image dataset. The Chinese captions of Flickr30K are obtained by translating the original English sentences using Google Translation API.

For word-level method, we tokenized each Chinese sentences into a list of words, e.g.

两个 朋友 享受 一起 度过 的 时 间

For char-level method, we split each Chinese captions into a list of Chinese characters, e.g.

两个 朋友 享受 一起 度过 的 时 间

In both method, the word or character is one hot encoded as a vector. Each image is transformed into a 4096D vector — the last layer of VGG net as illustrated in Figure 1.

RNN Models  
Our RNN model is the same with the one used in [2], which takes the input image at the first step together with a special start word token. It models the sequence generation by iterating through each word or char sequence.

The RNN is trained to combine a word, the previous context, to predict the next word as shown in Figure 2. We condition the RNN’s predictions on the image information via bias interactions on the first step, which means we feed image vector to the RNN only once.

The model was trained on a batch size of 100. Our RNN model is the same with the one used in [2], which takes the input image at the first step to predict the next word as shown in Figure 2. We condition the RNN’s predictions on the image information via bias interactions on the first step, which means we feed image vector to the RNN only once.

RNN Models  
Our RNN model is the same with the one used in [2], which takes the input image at the first step together with a special start word token. It models the sequence generation by iterating through each word or char sequence.

The RNN is trained to combine a word, the previous context, to predict the next word as shown in Figure 2. We condition the RNN’s predictions on the image information via bias interactions on the first step, which means we feed image vector to the RNN only once.

The model was trained on a batch size of 100. Our RNN model is the same with the one used in [2], which takes the input image at the first step together with a special start word token. It models the sequence generation by iterating through each word or char sequence.

The RNN is trained to combine a word, the previous context, to predict the next word as shown in Figure 2. We condition the RNN’s predictions on the image information via bias interactions on the first step, which means we feed image vector to the RNN only once.

The model was trained on a batch size of 100. Our RNN model is the same with the one used in [2], which takes the input image at the first step together with a special start word token. It models the sequence generation by iterating through each word or char sequence.

The RNN is trained to combine a word, the previous context, to predict the next word as shown in Figure 2. We condition the RNN’s predictions on the image information via bias interactions on the first step, which means we feed image vector to the RNN only once.

The model was trained on a batch size of 100. Our RNN model is the same with the one used in [2], which takes the input image at the first step together with a special start word token. It models the sequence generation by iterating through each word or char sequence.

Figure 1: Image vector produced by VGG net

Figure 2: Illustration of RNN sequence generation process

Result & Analysis  
The model generates sensible Chinese descriptions on 1,000 test images. In figure 3, we show two test examples generated with Beam size of 7. We can see that our model can recognize people and understand scenes and activity in which these objects are involved.

We also report the BLEUs on the Chinese sentences generated by the two methods with Beam size of 7. We compared the scores to those produced by [2] (FeiFei’s) in Table 1.

From the BLEU scores, we can see that the RNN model trained with char-level method for Chinese captions outperforms the word-level one. The former method performs very close to the original model trained on English sentences, with the word-level method performs slightly worse.

Thus, we came to the conclusion that this RNN model works universally well for image caption on different languages.

Limitations & Future Work  
The translation of the sentences were not so accurate, (both the training and testing sentences used Google translation) which made the the output not very accurate in the results. So what we could do, if we had enough time to work on, instead of using the sentences translated by Google, we may review on a few hundreds of images and manually correct them by ourselves. With that small set of clean data, we may try to train on that to see if it could work better.

Key References  
[2] Andrej Karpathy, Li Fei-Fei; Deep visual-semantic alignments for generating image descriptions; CVPR, 2015, pp. 3128-3137