A PARALLEL-FUSION RNN-LSTM ARCHITECTURE FOR IMAGE CAPTION GENERATION

Minsi Wang[*†], *Li Song*[*†], *Xiaokang Yang*[*†], *Chuanfei Luo*[‡]

[*] Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[†] Cooperative Medianet Innovation Center, Shanghai, China
[‡] Shanghai Research Institute of China Telecom
Email: mswang1994@gmail.com, {song_li, xkyang}@sjtu.edu.cn, luocf@sttri.com.cn

ABSTRACT

The models based on deep convolutional networks and recurrent neural networks have dominated in recent image caption generation tasks. Performance and complexity are still eternal topic. Inspired by recent work, by combining the advantages of simple RNN and LSTM, we present a novel parallel-fusion RNN-LSTM architecture, which obtains better results than a dominated one and improves the efficiency as well. The proposed approach divides the hidden units of RNN into several same-size parts, and lets them work in parallel. Then, we merge their outputs with corresponding ratios to generate final results. Moreover, these units can be different types of RNNs, for instance, a simple RNN and a LSTM. By training normally using *NeuralTalk*[1] platform on Flickr8k dataset, without additional training data, we get better results than that of dominated structure and particularly, the proposed model surpass *GoogleNIC* in image caption generation.

Index Terms— Image captioning, deep neural network, RNN, LSTM

1. INTRODUCTION

Image caption generation is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. Automatically describing the content of an image using properly formed English sentences is a very challenging task.

This task is even harder than the well-studied image classification or object recognition tasks, for which people challenge and then achieve breakthrough in *Large Scale Visual Recognition Challenge* (ILSVRC)[1]. Indeed, to generate exact sentences, the model should not only detect the objects of interest contained in the image, but also analyze the relationship between these objects. In addition, the weakness of computing capacity restrict the success of complex models. Fortunately, thanks to the rapid development of computer vision and natural language processing technologies, with the application of *Convolutional Neural Networks* (CNNs) and *Recurrent Neural Networks* (RNNs), recent works have made momentous progress, and present a unified method which dominates in image caption generation.

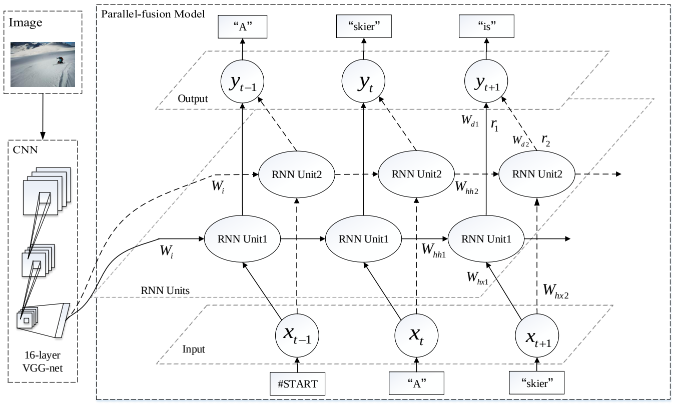

Fig. 1. The structure of the proposed Model

The first approach to use neural networks for caption generation was Kiros et al.[2], who proposed a multimodal Log-Bilinear model. But most of other recent works are different from it, which replace a feed-forward neural language model with a recurrent one. These works have some common key structures, and accordingly, we call those structures simply as dominated model or general model. In details, two of major parts, i.e., the CNN and RNN, play core roles in general model respectively.

Especially, for Flickr8k dataset, Mao et al.[3] present a *multimodal Recurrent Neural Network* (m-RNN) model which contains a *VGG-net CNN*[4] and a simple RNN. Karpathy&Li[5] gain the similar BLEU scores as Mao on Flickr8k in this task. Besides, Vinyals et al.[6] use LSTM instead of other RNNs in their model and unlike [3], [6] wisely show the image to RNNs at the beginning, leading to performance improvement finally. The temporary winner (Xu et al.[7]) archives the state-of-art performance. In this

[1] https://github.com/karpathy/neuraltalk

work, it presents an attention-based model and uses the CNN feature extracted from fourth layers, which increases the computational cost. However, as the amount of training data increases, the model with less training time will be needed.

Inspired by recent works, we observe the distinct performance between LSTM and simple RNN in BLEU[8] and *perplexity*, which motivates us to combine the advantages of them to present a parallel-fusion RNN-LSTM architecture (see Fig.1). For simple RNN, the length of generated sentence tends to be short and incomplete, but that of LSTM is totally different. It indicates that the simple RNN model does not generate length-wise sentence sometimes, which is partly because of its strong memory capacity that makes the sentence more likely to be end in advance. With no brevity penalty, short sentences decrease the error rate of prediction and lead to the higher BLEU performance. Additionally, considering the high Meteor performance of LSTM, we combine RNN and LSTM to generate a model with high Meteor (low perplexity) and BLEU performance.

Unlike [7] which contains additional training data and attached structures, our model only replaces the RNN part of the dominated model by combined parallel-fusion structures, and then aggregates their outputs with corresponding ratios to generate final results. With normal training procedures, the proposed model, which requires less parameters and training runtime, surpasses GoogleNIC benchmark [6] and performs at the same level of [7].

2. MODEL

The ultimate goal of our model is to improve performance in condition of promoting efficiency. Several recent works have shown that the dominated approach is universal and effective for image interpretation, which has powerful capacity in aligning visual and language data.

In this section, we propose a parallel-fusion RNN-LSTM architecture that contains two major structures without additional parts compared to the general model. The part of image representation is based on CNN while the part of caption generation is based on RNN structures. We apply them to extract image features and align visual and language data respectively. The proposed parallel-fusion model is showed in Fig.1.

2.1. Image and Sentence Representation

Following the method in [5], we represent the image as a single feature vector which is 4096-dimensional activations extracted just before the classifier from the fully connected layer of a pre-trained convolutional network on ImageNet. The pretrained CNN applied in our model is the 16-layer version of VGG-net[4], which is available online.

On the sentence representation, by calculating the amount of different kinds of words, we generate a dictionary of total words in dataset including the dot. And then, we represent each word as a one-hot vector[6] of dimension equal to the size of the dictionary. Especially, it should be emphasized that the size of word encoding space, the size of image encoding space and the size of hidden layers of RNN units should be equal to each other.

As we can see from Fig.1, the dominated model takes images and corresponding descriptions as inputs which are selected from dataset Flickr8k. Before fed into the CNN, images must be warped to a fixed size and be reshaped to the same dimension. Then the feature vectors of images and sentences could be embedded into the same space, and then are imported into the parallel-fusion image caption generator.

2.2. Parallel-fusion RNN-LSTM Generator

Based on the dominated model, our approach employs a novel parallel-fusion architecture of RNNs and reaches the goal of decreasing complexity and increasing performance. The details of the proposed models are described below (see Fig.1).

Our strategy of altering model divides the hidden layer into two parts and these two parts stay uncorrelated until the output unit. In forward-propagation process, those hidden layers receive the same feature vectors from source data and accept respective outputs of hidden layers from previous time, and then transmit outputs of RNN unit to y_t (see Eq.(3)) with corresponding ratios. We adopt fully recurrent network as example to illustrate our method. The corresponding formulae are shown as follows:

$$h_{1_t} = max(W_{hx1}x_t + W_{hh1}h_{1_{t-1}} + b_{h1}, 0) \tag{1}$$

$$h_{2_t} = max(W_{hx2}x_t + W_{hh2}h_{2_{t-1}} + b_{h2}, 0) \tag{2}$$

$$y_t = softmax(r_1 W_{d1} h_{1_t} + r_2 W_{d2} h_{2_t} + b_d) \tag{3}$$

$$dy_1 = r_1 \times dy \tag{4}$$

$$dy_2 = r_2 \times dy \tag{5}$$

where $\{h_1, h_2\}$ are the hidden units and $\{W_{hh1}$, W_{hh2}, b_{h1}, b_{h2}, W_{d1}, W_{d2}, $b_d\}$ are the weight parameters to be learned. Here, matrix dy is softmax derivatives. r_1 and r_2 are ratios we mentioned above. We multiple gradients of softmax derivatives by the same ratio $\{r_1,\ r_2\}$ for corresponding parts of hidden layers to maintain the balance of gradient update for each hidden unit.

Our method is different from the method of model ensemble which trains models respectively and merges their outputs for prediction. We merge their structures and train them together (see Eq.(1-5)). As shown in Fig.1, RNN unit1 and RNN unit2 are the same type of RNNs, and they could also be different. For instance, they may be a LSTM or a simple RNN unit. From the first two rows in Table.1, we observe that simple RNN archives higher BLEU scores and higher *perplexity* (low Meteor) than LSTM. That inspires us to combine their complimentary advantages together and realize the model which maintains constant perplexity and accomplishes higher BLEU scores. Therefore, a novel structure is proposed through replacing RNN unit1 and RNN unit2 by simple RNN and LSTM unit respectively in Fig.1. Moreover, we attempt to parallel more than two units and apply four LSTM units work together for comparison.

The LSTM unit used in our method is similar to reference [9] except showing the image to the units at the beginning. Besides, our strategy does not change the training methods and the models are trained in normal procedures by adjusting learning rates only. Lastly, for prediction, we do the forward-propagating and merge the outputs of hidden layers to get the results.

3. EXPERIMENTS

Next, we describe the dataset used for training and testing, followed by the evaluation of our approach for parallel-fusion structure and quantitative results which validate the effectiveness of our model. In the end, we compare the proposed model with recent works and discuss the contribution of our approach.

3.1. The Implementation Details

We use *NeuralTalk* released by Karpathy et al.[5] as our experimental platform. The training dataset Flickr8k contains 8000 images and each is annotated with 5 sentences. We use 6000 images for training, 1000 images for testing and the rest for validation, whose features are extracted by the 16-layer VGG-net using *Caffe*[10]. As in [5], we filter words to those that occur at least 5 times in the training set, which results in 2538 words for Flickr8k. We train models with different hidden sizes and different types of RNN units on a computer with *Intel*^{R} *Xeon(R) E5-2670 CPU*.

3.2. Evaluation

For quantitative evaluation, three different metrics are used to measure sentence generation process, which are *perplexity* (*PPL*), BLEU[8] and Meteor[11].

The perplexity is a standard measure for evaluating language model and can be described below [2][3]:

$$\log_2 PPL(w_{1:L}|I) = -\frac{1}{L}\sum_{i=1}^{L}\log_2 P(w_n|w_{1:n-1}, I) \qquad (6)$$

where L is the length of sentence. $P(w_n|w_{1:n-1}, I)$ represents the probability of generating the word w_n given the input image I and previous word $w_{1:n-1}$. $PPL(w_{1:L}|I)$ is the perplexity of sentence $w_{1:L}$ given the input I. The perplexity measures the uncertainty of the language model and a lower perplexity indicates a better score.

Besides perplexity, BLEU score has been the most commonly used metric so far in image captioning and already realized in *NeuralTalk*. Significantly, according to Karpathy[2], the BLEU evaluation script is exactly what's being used by Vinyals et al.[6] and Kiros et al.[12], with **no brevity penalty**, because most of the generated sentences are not too short, so in practice the $BP \approx 1$. However, there has been criticism of BLEU, for a more convincing evaluation, so we report another common metric Meteor[11] in addition.

Table 1. BLEU-1/PPL/Meteor metrics compared to other methods on Flickr8k dataset

Model	B-1	*PPL*	Meteor	Hyp_len
sRNN-h512	66.7	27.58	16.53	7271
LSTM-h512	60.5	**15.58**	16.93	7806
2LSTM-h128	60.1	18.80	17.03	8497
2LSTM-h256	61.3	18.21	17.80	8651
4LSTM-h128	58.4	17.29	17.98	**9648**
Mix4v6	61.6	20.70	17.13	8633
Mix5v5	63.1	19.76	18.26	8834
Mix6v4	**64.7**	19.68	**18.85**	8550
Mix7v3	64.6	17.76	18.20	7937
Mix8v2	61.7	18.33	16.58	7514
GoogleNIC[6]	63	-	-	-
Log Bilinear[2]	65.6	-	17.31	-
Soft-Attention[7]	67	-	18.93	-
Hard-Attention[7]	67	-	20.30	-

As for efficiency evaluation, we take model size and training runtime into account. The model size corresponds to the amount of all the weights we need to learn. The training time measured by system clock is another fatal criterion and it does not include the validating time during training. Furthermore, we introduce candidate length (hypothesis length) which is related to BLEU metric for qualitative evaluation.

Moreover, a few options exist for comparison. Similar to the discussion in [7], the first one is the difference in choice of convolutional feature extractor. We take GoogLeNet and Oxford VGG features as comparable features and compare corresponding results directly. The second is the difference between final features. Some of recent works only generate CNN features, but Xu et al.[7] generate attention features additionally. We just take it as a reference and compare whenever possible. The final is the difference between source data splits. To achieve fair comparison, we use the same publicly predefined splits as in the previous work[5].

When evaluating models, **BeamSearch** method has been applied and it iteratively considers the set of k best sentences up to time t as candidates to generate sentences of size $t+1$, and keep only the resultant best k of them. Besides, the beam size applied in experiments is 20.

3.3. Caption Generation Results

We provide a summary of the experiments and report main generation results on the Flickr8k dataset in Table 1. Our model surpasses *GoogleNIC*[6] which archives the temporary

[2] https://github.com/karpathy/neuraltalk/pull/5

state-of-the-art BLEU performance in the last year under the same generation structure, and we obtain comparable performance in Meteor with *soft-attention* method[7].

Table 1 shows the results of diverse parallel-fusion models. **sRNN-h512** represents a simple RNN model in which the hidden size is 512. And the *Hyp_len* is the candidate length. *GoogleNIC* is a model realized by Vinyals et al.[6]. Besides, the hidden size is 256 in all mix-models.

Above all, it is noted that the model **LSTM-h512** is a attempt to reproduce Google's LSTM results, which is already realized in *NeuralTalk*, so all settings are as described in Google paper[6], except the VGG-net is used for CNN features instead of GoogLeNet. We test all these models and validate that their generated captions are all in high quality except sRNN which generates obviously briefer sentence than others. This leads to high BLEU scores because of no brevity penalty. So the result of single sRNN will not be token into account in further discussion.

As above mentioned, there are two parallel-fusion strategies in our experiments. The first type of novel parallel-fusion models replaces RNN units with only LSTM units. The BLEU scores of them are approximately equal to that of LSTM, and perplexity are a little bit higher than that of reference, but their performance in Meteor shows a different result. Especially, the model that we apply four LSTM units in parallel has unique length and the highest Meteor among these LSTM only models. The results of this strategy imply that the combination of only LSTM units results in no remarkable improvement in performance but just increases the amount of generated words.

The second type combines LSTM units and sRNN units, merges their outputs and lets them work together. The model **Mix3v7** represents a structure in which the outputs of LSTM units and sRNN units are multiplied by mix-factors 0.3 and 0.7 respectively. It is encouraging that all unit-mixed models gain satisfactory performance. Particularly, as we can see from Table 1, the models **Mix6v4** and **Mix7v3** obtain comparable results and both surpass *GoogleNIC*[6] in BLEU in the same benchmark without any additional training data and processing structures. Besides, both of them beat *LogBilinear*[2] method and archive comparable performance with *soft-attention*[7] in Meteor.

3.4. Efficiency Discussion

The proposed model seems to be more complex than the dominated one we mentioned, because it looks containing more units and more amount of variables and seems to take more time on calculation. However, the fact is that our approach obtains the approximate performance but only needs half of origin size of hidden layers. Now that hidden size becomes a half of origin size, so the computational runtime is supposed to decrease to some extent. The final results of *Mix6v4* and *Mix7v3* show that our parallel-fusion strategy not only decreases the complexity of model but also promotes a little bit performance.

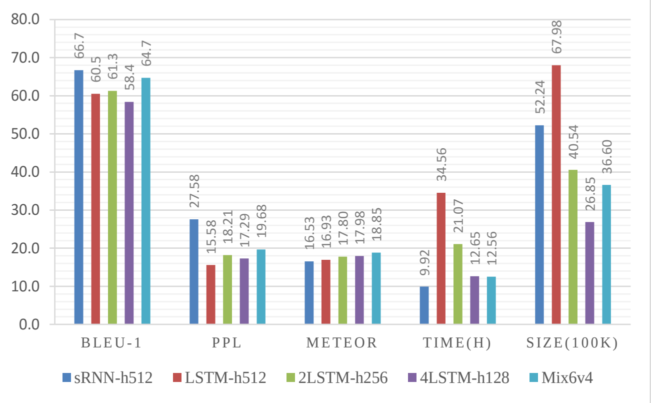

Fig. 2. The evaluation results. The time is the training runtime. For comparable performance, mixed models require less memory and run faster than single-LSTM model.

In addition, we take other evaluation criteria into account for quantitative analysis. The evaluation results of training time and model size are presented in Fig.2. Considering both performance and efficiency, we find that the model *Mix6v4* is in a good trade-off of them. Fair perplexity, medial training time and medial size make it become a unique choice. When obtaining equal performance to LSTM, almost novel models spend less time and occupy less memory than the dominated ones. The size of whole LSTM structures is about 6.8M and it is trained with 9000 batches which takes near 35 hours. As we can see, our parallel-fusion models which spend less than half of resources are still able to improve performance.

4. CONCLUSION

In this paper, we introduce a novel parallel-fusion RNN-LSTM architecture for image caption generation. The proposed approach makes a remarkable breakthrough in performance and efficiency. Without any additional training data and structures, our parallel-fusion models can beat *GoogleNIC* in BLEU and *LogBilinear* in Meteor on Flickr8k dataset. In other respects, our models spend less than half of resources occupied by the dominated model while improving performance. For future works, we intend to explore the limitation of the amount of parallel threads and use even more complex image features to boost performance further.

Acknowledgement: This work was supported by NSFC (61527804,61521062), 863 project (2012AA011703), the 111 Project (B07022 and Sheitc No.150633) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

5. REFERENCES

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, pp. 1–42, 2014.

[2] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel, "Multimodal neural language models," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 595–603.

[3] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *ICLR*, 2015.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] Andrej Karpathy and Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[6] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[7] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2048–2057.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[9] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[11] Michael Denkowski and Alon Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014, vol. 6.

[12] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *CoRR*, vol. abs/1411.2539, 2014.