(The report is not 100% finished.)

Which Settings Give Better Results?

Xi Chen, University of Gothenburg

Abstract

In this project, an experimental study in image captioning in Chinese language is carried out. Totally twelve models were trained with different settings (such as both applying and not applying Chinese text segmentation, using both pre-trained VGG19 and ResNet101, both with and without fine-tuning, etc.) on both Flickr8k-CN and Flickr30k-CN datasets, and twenty-four checkpoints were examined. The trained models/checkpoints were then evaluated by BLEU, ROUGE_L and CIDEr criterias for comparison and to see how they performed. Evaluation results were shown in tables, and were discussed.

1. Introduction

Image caption as a significant part of artificial intelligence has been deeply researched by many researchers in the past decade. However, most of the research was based on image captioning in English language, and most of the databases for research that can be found are also English-based. Chinese is the world's most spoken language, but related research in Chinese language in the field of natural language processing and computer vision, especially in image captioning still cannot compete with those in English language. Some researchers have realized this and have done several pilot research on this topic. Inspired by their work, this little project was conducted in order to do some experiments on image captioning in Chinese language.

1.1 Related works

The authors of [1] have extended the popular Flickr8k with both machine-translated and human-annotated Chinese captions and experimented how they performed in image captioning in Chinese, and concluded that "Baidu translation is preferred to Google translation", and "which model is more suited for Chinese captioning depends on what ground truth is used" and the NIC model performed in image captioning in Chinese just as well as in image captioning in English. H. Peng & N. Li [2] have in their experiment showed that according to the BLEU score the char-level method worked better than the word-based method in generating Chinese captions for Flickr30K images. However, the ground truth Chinese captions they used are directly translated from English captions in the Flickr30K database by Google Translation API. In the work [3], the authors has

extended the popular Flickr30k database with machine-translated Chinese captions and then proposed a model which can estimate the fluency of the machine-translated sentences in order to decide if they should be used in the image captioning model, or their importance should be decreased because of low fluency when fed into the image captioning model, which provides us a way to use machine-translated ground truth in image captioning, and at the same time helps to increase the performance of the image captioning model by excluding bad machine-translations.

1.2 Goal

The goal of this project is not to propose any model for image captioning in Chinese which can compete with the state-of-the art researches, but rather, based on the code of [4], conduct several experiments on generating Chinese captions for both Flickr8K and Flickr30K images with ground truth captions from Flickr8K-CN and Flickr30K-CN, using trained models with different settings; and to finally evaluate their performance using BLEU scores when different beam size are chosen.

By “models with different settings”, it is meant that models using both char-based method and segmentation-based models, using both pre-trained Resnet101 and VGG19, both without fine-tune and with fine-tune, both with best BLEU-4 scores and with early stop.

1.3 Data

The data used in this project is Flickr8k-CN and Flickr30k-CN. [5] Flickr8k-CN is a dataset of images with Chinese captions extended by [1] from the popular Flickr8k dataset. Flickr30k-CN is a dataset extended from Flickr30k by [3] which contains images with Chinese captions; however, all the captions in the train set and val set are machine-translated, but human-translated in the test set. The splits of these two datasets are shown in the following table.

	Flickr8k-CN			Flickr30k-CN		
	train	val	test	train	val	test
Images	6000	1000	1000	29,783	1000	1000
Human-annotated Chinese sentences	30,000	5000	5000			
Machine-translated Chinese sentences	30,000	5000		148,915	5000	
Human-translated Chinese sentences			5000			5000

2. Methods

The methods used for this project will be explained in the following parts: the code for programming, the model, the different experiments settings and the methods used for evaluation of the results.

2.1 Code

The code of this project is mainly based on [6]. It is a tutorial code to image captioning using pytorch [7]. The image captioning model used in this tutorial is based on the paper [8], which to some extent no longer is the state-of-the-art model, but still suitable for pilot studies. The code for this project is available at:
https://github.com/guschenxi/aics-project.

Some files have been added and many files in this tutorial have been rewritten to suit for this project. “create_json_flickr30kcn.ipynb” and “create_json_flickr30kcn.py” were added in order to transfer the data from Flickr30k-CN and Flickr8k-CN into the required form which can be used as the input data into the model. Minor changes have been made in “model.py” and “train.py” in order to make them fit the experiment settings. Greedy search and beam search functions have been added into “utils.py”. “checkpoints.py” was added to store all the names of the trained models with different settings. “eval.py” was re-written so that it was able to conduct evaluation for all models with different settings, with beam size from 1 to 5. “caption_greedy.ipynb” and “caption_beam.ipynb” were added to perform image captioning in Chinese for raw images using greedy search and beam search.

2.2 Model

The model for this project is based on the paper [8]. It consists of an encoder and decoder with an attention mechanism. The encoder takes in image inputs into pre-trained Convolutional Neural Networks and encodes them into smaller representations of learned important features from the images.

Since image captions are tokens with sequence, the decoder is therefore an Recurrent Neural Network, here in this project, an LSTM (Long Short-Term Memory) which takes a look at the encoded images and generates a caption of the image word by word.

The attention mechanism enables the encoder to look at the corresponding part of the image when generating each token, i.e. the pixels of the more important part of the image corresponding to the token get bigger weights.

2.3 Experiment Settings

2.3.1 Chinese Text Segmentation

Different to English and most other languages, Chinese is a character-based language rather than an alphabet and word-based language. Moreover there are no spaces between characters (or character groups which can have a separate semantic meaning). Therefore, tokenization for Chinese language processing is not as simple as in English which can be done by splitting a sentence by spaces. However, there are some popular tools which provide Chinese text segmentation, such as “Jieba” [9]. The Flickr8k-CN and Flickr30k-CN databases used in this project were already tokenized, so that there are added spaces between each semantic unit (either character or character group). In this project, both character-based models (by taking away the added spaces, and splitting the sentence again by characters) and token-based (segmentation-based) models were trained and experimented, to see how well they perform.

2.3.2 ResNet101 and VGG19

With the rapid development of computer vision, more researchers have presented deeper and deeper neural networks in order to deal with more complex image-related problems. VGG and ResNet were two of those presented deeper neural networks which are widely used nowadays. [10]

VGG (stands for Visual Geometry Group, a group of researchers at Oxford who developed this architecture [11]) is a convolutional neural network structure presented in 2014 by Simonyan and Zisserman. [12] According to their paper, the model achieves 92.7% top-5 test accuracy in ImageNet. VGG19 is a 19 layer deep variant of VGG networks.

ResNet, short form of Residual Network is another neural network structure presented in 2015 by K. He et al. [13] ResNet has similar structure as VGG, but can better deal with the problem of vanishing gradients. ResNet has better performance which achieves 93.3% top-5 test accuracy on ImageNet, than VGG, and works faster than VGG. ResNet101 is a Residual Network variant which is 101 layer deep.

Pre-trained models that are trained on large databases and contain feature representations of the data they are trained on can be used to different data in order to save a great amount of time. The learned features are transferable and can benefit other models. The pre-trained models of VGG19 and ResNet101 are provided by most deep learning APIs, as well as Pytorch.

In this project, both pre-trained VGG19 and pre-trained ResNet101 were experimented in order to examine their performance.

2.3.3 Feature Extraction and Fine Tune

Using pre-trained models which contain transferable learned features to train new models is considered as transfer learning. This brings out two approaches to transfer learning: feature extraction and fine tuning.[14] Feature extraction means that the

parameters of the pre-trained model are not changeable, and the feature representations learned will be used directly in the new model; while fine tuning allows the weights in the pre-trained model to be changed and fine-tuned, so that it will be adapted to and benefits the new task. In order to see how allowing fine tuning will affect the performance of the models, experiments both without fine tuning (feature extraction) and with fine tuning were conducted in this project.

2.3.4 Models with and without Best BLEU Scores

BLEU (bilingual evaluation understudy) is an approach brought up by Kishore Papineni, et al. in 2012, which originally was for evaluating the quality of machine translation. The BLEU score is to calculate how many n-grams in the candidate translation match the n-grams in the reference translation. The calculated BLEU score is a number between 0 and 1, which shows how close a machine translation is to a professional human translation. [15] BLEU-1, BLEU-2, BLEU-3 and BLEU-4 refer to the cumulative 1-gram, 2-gram, 3-gram and 4-gram BLEU scores. Even though BLEU was brought up to evaluate machine translation, it is nowadays broadly used to evaluate image captioning.[adding chinese captions to images] Therefore, BLEU scores were also used in this project to evaluate the performance of different models.

The tutorial code provided by [4] has a setting which saves the checkpoint after each epoch, and replaces it with a new checkpoint after a new epoch is finished. Therefore, the checkpoint after the last epoch is always saved. It also calculates the BLEU-4 score after each epoch, and if the BLEU-4 score has increased compared with it from the previous epoch, the checkpoint will be saved (or replace the previous one) and named with “BEST”. In the tutorial code, it has a mechanism which enables the training process with early stop when the BLEU score doesn't increase after 20 epochs. When training stops, the last checkpoint will replace the previous one and will be used as one of the final models for this project. The 20th checkpoint before this last checkpoint will of course be the one with the best BLEU-4 score and be saved and named with “BEST”, which will also be used as one of the final models. Each of the settings mentioned above got two checkpoints after training, one with best BLEU-4 score and one achieved 20 epochs after the one with best BLEU-4 score.

2.4 Training

Training epochs were set to 100, however, early stopping were triggered in all training processes. The learning rate for the encoder was set to 1e-4, only when fine tuning was turned on. The learning rate for the decoder was set to 4e-4. The dimension of word embeddings and of the attention layers was set to 512. The hyperparameters used for training were the same as the tutorial code, remaining not changed. The only difference is that when conducting experiments on using pre-trained VGG19, the dimensions have been changed to adapt VGG19’s structure. The whole training process will not be explained in detail here, since no significant change has been made compared to the tutorial.

After training using 6 different experiment settings shown in the following table, on both Flickr8k-CN and Flickr30k-CN, saving two checkpoints (explained above in 2.3.4) for each, totally 12 models with 6 different settings were trained and 24 checkpoints were saved.

	Character-based	Segmentation-based	VGG19	ResNet101	Feature Extraction	Fine-tuning
1	x			x	x	
2		x		x	x	
3	x			x		x
4		x		x		x
5	x		x		x	
6		x	x		x	

2.5 Evaluation

The criterias used for evaluation are BLEU, ROUGE_L and CIDEr, same as many research have done. [1,2,3]

From the beginning, the BLEU scores were calculated using the “corpus_bleu” function inside nltk.translate toolkit [16]. Then an evaluation code for natural language generation which automatically conducts evaluation with different criterias was found at [17]. However, some minor changes have been made to fit this project and some small bugs in the code were also fixed.

2.5.1 BLEU score

2.5.2 ROUGE_L

2.5.3 CIDEr

2.5.4 Beam search

3. Results

The results of the evaluation of the trained models are shown in tables in the appendix attached below this report. Some of the experiment settings are written in short forms. “BB” refers to the models with **B**est **B**LEU scores as described in section 2.3.4; while “NB” (**N**ot **B**est BLEU) refers to the models (checkpoints) trained 20 epochs more than “BB” models. “8k” and “30k” refers to models trained on Flickr**8k**-CN and Flickr**30k**-CN. “FE” and “FT” refers to models trained without

fine-tuning (**F**eature **E**xtraction) and with **F**ine-**T**uning. “RN” and “VG” refers to models trained using pre-trained **R**es**N**et101 and **VG**G19. “b1”-“b5” means evaluation results using different **b**eam sizes from 1 to 5. “B-1”, “B-2”, “B-3”, “B-4” refers to **B**LEU**-1**, **B**LEU**-2**, **BL**EU**-3** and **B**LEU**-4** scores.

Generally, it can be seen that models trained on Flickr8k-CN got much better scores than models trained on Flickr30k-CN. One possible reason is that the ground truth captions in the train and val split of Flickr30k-CN were machine-translated, some of which lack fluency.[3] In the following of this chapter, evaluation results are shown depending on the choices of seam size and the settings of the models.

3.1 Which Beam Size Performs Better?

The results (Result 1A & 1B) do not show a fixed pattern about which beam size generates sentences with higher scores. The performance depends on the settings of the model, i.e. which dataset it used, if fine-tuning was turned on or not, which pre-trained model was used in the encoder, etc. However, in most cases, beam size between 3 and 5 gave higher scores.

3.2 Character-Based vs Segmentation-Based

As can be seen in the table (Result 2), except three exception scores, almost all character-based models have gotten higher scores than those with exactly the same settings but segmentation-based models. This also proves [2]’s previous finding.

3.3 ResNet101 vs VGG19

As the results (Result 3) in the table show, most models using pre-trained ResNet101 have achieved better scores than those with exactly the same settings but using pre-trained VGG19. However, one exception is the model setting marked as “BB-30k-seg-FE”, in which all scores from the model using VGG19 beaten that using ResNet101. (Probably because of training errors. The model is being re-training at the moment, hopefully the scores would be changed.)

3.4 Feature Extraction vs Fine-Tuning

The results (Result 4) in the first table indicate that, for almost all of the models marked by “BB”, fine-tuning the encoder gave better BLEU and ROUGE_L scores, than only using the encoder for feature extraction. On the contrast, the situation changed when it came to models marked by “NB” (models trained for 20 more epochs): not fine-tuning the encoder gave better BLEU, ROUGE_L and CIDEr scores for most models than those using fine-tuning.

3.5 Checkpoints with Best BLEU Score vs Continue Training

(Result 5) A large percent of the scores (with several exceptions) show that models marked by “BB” achieved higher scores than those with exactly the same setting but marked by “NB”, which means that most of the models trained for another 20 epochs do not perform better than the checkpoints saved 20 epochs earlier. It can be concluded that models trained for more epochs do not always perform better, but can lead to overfitting and therefore worse performance. Early-stop when BLEU-4 score does not

increase more, as the tutorial code programmed, seems to be a good approach. However, other criterias other than BLEU remain to be tested in this context.

4. Discussion

4.1 Criticism Bleu score

4.2 Training data bias

The train set and val set in Flickr30k-CN consists of machine-translated Chinese sentences from English, while the test set consists of human-translated sentences, which was probably the reason why models trained on Flickr30k-CN got much lower scores, compared with those trained on FLickr8k-CN which consists of only human-translated and human-annotated sentences.

4.3 Gender bias

4.4 Interrupted Training

Several training sessions were terminated in the middle of the training, because of bad Internet connection. Even though the tutorial code provides with the function which enables the checkpoints to be saved and restored from the middle, it seemed according to the evaluation scores that errors have occurred anyway which affected the final comparison among the models. However, these checkpoints which have terminated in the middle have been re-trained, and the evaluation scores have been corrected before analysis. In future works, interrupted training sessions should be re-trained from beginning rather than restored from saved checkpoints.

5. Conclusions and Further Work

Train on MS COCO

Evaluation with other standards

References: (not finished)

[1] [X. Li et al.]
[2] [H. Peng and N. Li]
[3] [W. Lan et al.]
[4] https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
[5] https://github.com/li-xirong/cross-lingual-cap
[6] https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
[7] https://pytorch.org/

[8] *Show, Attend, and Tell*
[9] Jieba
[10] https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96
[11] https://towardsdatascience.com/vggnet-vs-resnet-924e9573ca5c
[12] Very Deep Convolutional Networks for Large Scale Image Recognition
[13] Deep Residual Learning for Image Recognition
[14] To Tune or Not to Tune?
[15] https://en.wikipedia.org/wiki/BLEU
[16] https://www.nltk.org/api/nltk.translate.html
[17] https://github.com/Maluuba/nlg-eval