



中山大學
SUN YAT-SEN UNIVERSITY

深度学习

Deep Learning

题目 Title: AI Meme Generator Based on Show and Tell Model

院系
School (Department): 智能工程学院

专业
Major: 智能科学与技术

学生姓名
Student Name: 李睿能 卢晴 马小涵 张翠宁

学号
Student No.: 18364047 18364065 18364070 18364117

时间: 2020 年 1 月 18 日

Date: Mon January Day 18 Year 2020

AI Meme Generator Based on Show and Tell Model

Li Ruineng, Lu Qing, Ma Xiaohan, Zhang Cuining

(School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510000, China)

Abstract: We construct an AI meme generator in our project which can produce a relevant and humorous caption on the basis of a given image. A ResNet network is used to capture image embeddings. And the outputs of the CNNs are passed to an attention-based LSTM model that generates captions. We use an efficient ResNet convolutional neural network to raise accuracy. Furthermore, Luong attention mechanism is used in our project to perform attention on the LSTM cell. To find a high-efficiency and appropriate model to generate word vectors, We compare BERT, word2vec and GloVe models to find the best one and apply it in our work. In addition, we design a loss function as an index of humor based on humor recognition theory.

Keywords: attention mechanism; humor recognition; show and tell model; meme generator

1 Introduction

Memes are symbols that combine pictures and words to express specific phenomena, hot topics and specific emotions, and represent contemporary thoughts and cultures. As a kind of network subculture, Internet meme has been promoted to the mainstream culture in recent two years and has even become one of the indispensable expressions in our daily life. As a non-verbal communication tool, Internet emoticons greatly facilitate people to exchange information and convey emotions, making interpersonal communication more vivid and diversified. For example, during the "two sessions" in 2017, there were many reports on memes in mainstream media.

Our research project—generating meme is similar to image captioning, including not only image feature recognition but also description of image features. Image caption is a popular research area of Artificial Intelligence that deals with image understanding and a language description for that image. Generating well-formed sentences require both syntactic and semantic understanding of the language. Therefore, our research will also contribute to the development of Image caption in some extent.

On the basis of image caption, we added humorous recognition. We want to bring humor to the human level by training a lot of datasets. In order to achieve this goal, datasets are the essence of this meme generator. The dataset consists of about 400,000 tagged and illustrated images. There are 2,600 unique image-tag pairs, retrieved through Python scripts from Memegenerator.net. A picture corresponds to a label, which is a simple description of the picture, and each picture is associated with many different illustrations (about 160).

The basic framework of the meme generator is an encoder-decoder picture generation system, which first emits CNN images and then generates text caption using an LSTM RNN. The goal of the encoder is to give a meaningful state for the decoder to start generating text. The project uses Inception-v3, pre-trained on ImageNet, as the encoder model and put the last layer of hidden CNN as the output of the encoder. When the meme template enters the Inception model, the output is a set of vectors, or image embedding, that reflect the content of the image. The image is then projected into the word embedding space to facilitate subsequent text generation. Three different encoder

models were tried. The simplest one only input images, the other input images and labels, and the last input was also images and labels, but with the attention mechanism.

This project main aims to generate a humorous caption in a manner that is relevant to the initially provided image. And we intend to explore how to train AI to recognize humor.

2 Background and Related Work

2.1 Image Captioning Models

The writers of *Show and tell: A neural image caption generator*[1] put forward a model for image captioning. In their project, the image captioning model contain an encoder-decoder scheme. The encoder is a deep convolutional neural network, CNN which makes use of images as input. And the output of the CNN is fixed-length vector embeddings that are then fed into decoder. The decoder begins with a trainable adequately connected layer which produces the initial state of the RNN network for caption generation. And the authors use a Long Short Term Memory (LSTM) network as a variant of RNNs[1].

Most image captioning models use the similar scheme above. Novel ideas like deep bi-directional LSTMs[2], attention mechanisms[3] and some stylized models such as StyleNet[4] are attempted to promote the performances of the image captioning models or produce stylized models such as romantic and humorous captions. In our project, we construct a humorous image captioning model to gain our stylized meme, and polish our model using the preceding works above.

2.2 Recurrent Neural Networks (RNNs) for Language Modeling Tasks

RNNs are widely used as a decoder in sequential NLP tasks, especially when the input data does not have a fixed size. (引用) Among different types of RNNs, Long Short-Term Memory(LSTM) is a good choice in this task due to the fact that it can remember data from long periods of time with the help of “gating mechanisms”. The common used LSTM cells are based on following equations:

$$\begin{aligned} i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\ f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\ o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{cx}x_t + W_{cm}m_{t-1}) \\ m_t &= o_t \circ c_t \\ p_{t+1} &= \text{softmax}(m_t) \end{aligned}$$

where i is the input gate, f is the forget gate, o is the output gate, W are the trainable matrices and m is the memory output. The output is the probability distribution for each word in the vocabulary.

2.3 Word Embedding

Using vector embeddings to represent words is a vital concept to capture semantic similarities in NLP tasks. We try two word-embedding methods, which are the GloVe trained on a very large corpus and the state-of-the-art pretrained model BERT. GloVe word embeddings constitute almost the most suitable choice of word representation for our project, based on the fact that they have been trained on very large corpora, offering an option of words from Wikipedia with 6 billion tokens. However, incapacity of presenting polysemy phenomenon and weaknesses of autoregressive model

are well-known shortage for GloVe. So we give a try on BERT, which is based on the official pretrained model and fine-tuned by the short-jokes dataset crawled from Reddit.

2.4 Attention Mechanisms for RNNs

In sequential NLP tasks, fixed-length vectors are often used to interpret the features of the input sequences. However, this may lead to a fact that some information will be lost if the input sequences is long. To solve this bottleneck problem, attention is introduced to deep learning. When human describing an image, we will pay more attention to things that are more closely related to the main object. Neural networks can also focus on part of a subset of the information they're given by using attention. The attention distribution is usually generated with content-based attention which allows the RNNs to look at different position of an image every step. One common variant of attention is introduced by Luong[5], we try to implement this attention mechanism in our model. The difference between traditional RNN and the RNN with attention is shown in Figure 1.

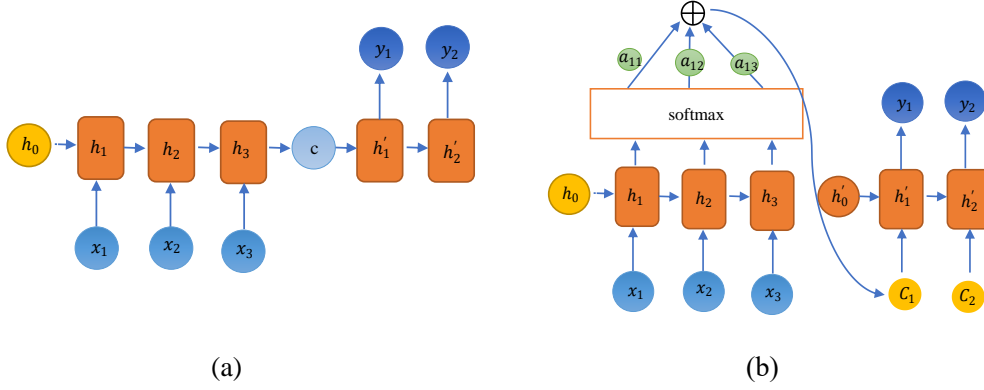


Figure 1: A visualization of traditional RNN(a) and attention-based RNN(b). In figure(a), h is the hidden state of Encoder, c is a vector that contains the information of the last hidden state of encoder and h' is the hidden state of decoder. In figure(b), c is the context vector, a is the attention vector.

3 Approach

3.1 Dataset

3.1.1 The reasons for choosing the MSCOCO

COCO is a large image dataset designed for object detection, segmentation, person key points detection, stuff segmentation, and caption generation. Cocos collect data by making extensive use of Amazon Mechanical Turk. COCO datasets now have three annotation types: object instances, object key points, and image captions, which are stored in JSON files. The main features of the MSCOCO dataset are as follows: (1) Object segmentation (2) Recognition in Context (3) Multiple objects per image (4) More than 300,000 images (5) More than 2 Million instances (6) 80 Object categories (7) 5 captions per image (8) Key points on 100000 people.

A meme contains multiple objects, so Object segmentation is needed. Moreover, the recognition of context in the picture is very important for the scene analysis of an meme, especially for the text caption of a meme. In addition, this dataset has More than 300,000 images, More than 2 Million instance and 80 object categories. And the fact that each picture has five captions is also

very much in line with the requirements of meme captions. This data set aims at scene understanding, which is mainly extracted from complex daily scenes. The location of the target in the image is demarcated through accurate segmentation. The image consists of 91 types of objects, 328,000 images, and 2,500,000 labels.

This data set mainly solves three problems: target detection, context relationship between targets, and accurate positioning of targets in 2 dimensions. Comparison diagram of data set:

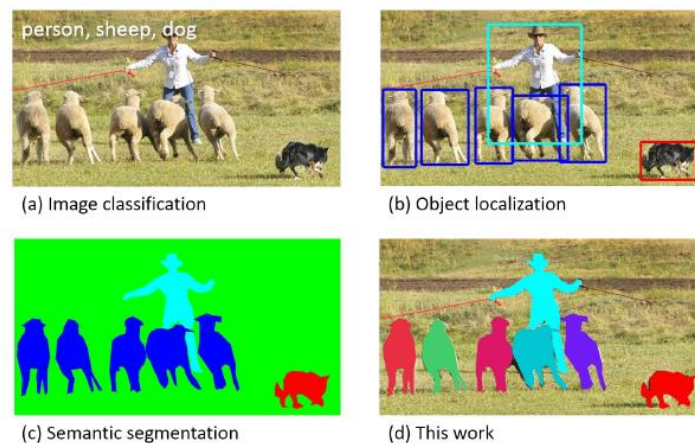


Fig. 1: While previous object recognition datasets have focused on (a) image classification, (b) object bounding box localization or (c) semantic pixel-level segmentation, we focus on (d) segmenting individual object instances. We introduce a large, richly-annotated dataset comprised of images depicting complex everyday scenes of common objects in their natural context.

Fig1.refer to the 《目标检测数据集 MSCOCO 简介》

3.1.2 Difficulties in creating datasets

Since the data is divided into images and image labels, the biggest difficulty in finding the data set is to download the images and the corresponding title text in batches. The pictures found on the meme generator are pictures containing text titles, and data separate from the pictures and copywriting is not directly found. So separating the memes' title text from the pictures is also a challenge.



Fig2.refer to the website <http://msvocds.blob.core.windows.net/coco2014>

3.1.3 The process of creating a dataset

First we download and process MSCOCO from <http://msvocds.blob.core.windows.net/coco2014> and <http://msvocds.blob.core.windows.net/annotations-1-0-3> and then we build dataset, the dataset includes image_id: integer MSCOCO image data: string containing JPEG encoded image in RGB color space. caption: list of strings containing the (tokenized) caption words caption_ids: list of integer ids corresponding to the caption words. This is the simple dataset.

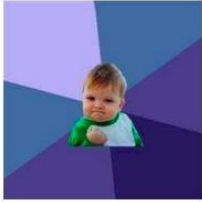

Image	Label	Caption
	<ul style="list-style-type: none"> • success kid 	<ul style="list-style-type: none"> • Didnt study for a test still get a higher grade than someone who did • Ate spaghetti with a white shirt on no stains • New neighbors Free Wifi ...
	<ul style="list-style-type: none"> • awkward seal 	<ul style="list-style-type: none"> • You laugh when your friend says something He was being serious • took a photo camera the wrong way • Goes to friends house Friend isn't there yet ...

Table 1: Sample dataset

3.2 Model Variants

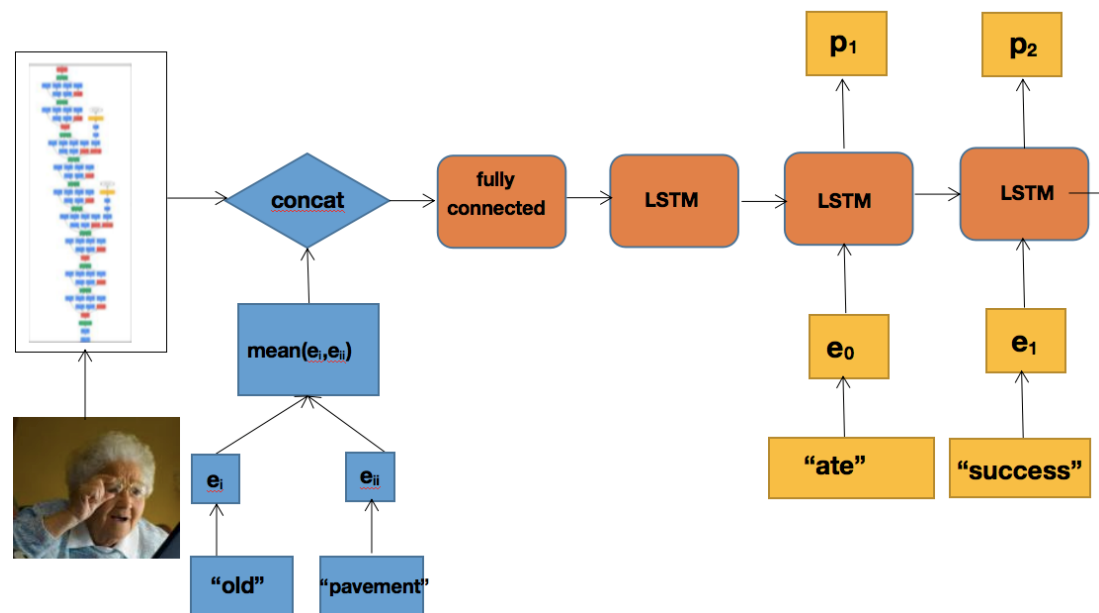


Figure 2: A visualization of our encoder-decoder model. e_i is word embeddings, p_j is the probability distribution obtained at each LSTM time-step.

3.2.1 Encoder

3.2.1.1 CNN

The encoder is in order to provide an initial state to the decoder to initiate the text generation process. We use the CNN ResNet model and take the last hidden layer of the CNN as the encoder output. The output of the ResNet model is a fixed length vector, image embedding, that captures the content of the image. Traditional ShowAndTell model relies on the CNN model Inception-v3 to capture the image embeddings[6].[7] In the paper we reference, they replace the CNN model Inception-v3 by the AlexNet in their actual operation. The motivation behind the operation is to reduce the time for the Inception-v3 model training to increase training efficiency. In our project, we rely on the CNN ResNet model instead of the AlexNet model and the Inception-v3 model. The ResNet model employs a residual learning framework to ease the training of networks that are considerably deeper than the previous which work out the problem that deeper neural networks are more difficult to train. Previous projects have demonstrated that these residual networks are easier to optimize, and can raise accuracy from tremendously increased depth.[8] So we conjecture that the ResNet model can gain a good result in the task and rely it in our project. We train the new ShowAndTell model and the results of our experiments to conclude whether the use of the CNN ResNet model is more efficient and better than the preceding projects.

3.2.1.2 Attention mechanism

After obtaining the image embeddings from CNN, we put them through a fully connected layer. Before the decoder, we add an LSTM network. This LSTM network takes the projected image embeddings as the initial state and runs the word embeddings of the labels through the LSTM. Luong attention mechanism is used here to perform attention on the LSTM cells as introduced in 2.4 in this paper. The output serves as the initial state of the decoder.

In our experiment, although we have tried many times to add Luong attention to the LSTM, we have encountered a lot of problems. For example, we first insert attention to the LSTMs of the decoder because we confuse the concepts of attention interfaces and the attention between RNNs. There are lots of works that focus on attention between RNNs such as tasks like parsing text which allows the model to glance at words that parse tree[9]. However, in this task, what we need is to generate a description of the image. So attention should be used to decide which part of the image has a larger contribution to the caption. Therefore, it should be used on the interface between a convolutional neural network and RNN.

3.2.2 Decoder

As is discussed in section 2.2, we choose LSTM as the decoder. Bi-directional LSTM first came to our minds. BiLSTM makes some adjustments to the traditional LSTM, which have an excellent performance on NLP tasks like translations and text sentiment classification. It takes into account the context which consists of both the previous and the latter information. However, in our experiment, we found that it might not be a good choice to use bi-directional LSTM as the decoder in this task. The reason is that the function of this neural network is to decode the embedding information and generate the caption from the vocabulary. Thus, when choosing a word to finish the sentence, what matters is the words before the target word. As for the words after the target, they have little influence to the target word. Therefore, using BiLSTM will be a loss of space and time in this kind of tasks. An unidirectional LSTM network was finally chosen as the decoder.

Our encoder-decoder model is based on Show and Tell Model[10]. One modification we made is that the InceptionV3 network was replaced by AlexNet and ResNet. Another one is the use of pretrained word embeddings discussed above rather than randomly initialized word vectors.

3.3 Loss Function

Most existing studies on humor recognition indicate a similar latent semantic structure behind humor in four aspects: Incongruity, Ambiguity, Interpersonal Effect and Phonetic Style. In our task, we try to employ a brand new loss function specific for humorous-sentence generation based on the humor linguistic theory.

To quantize incongruity in each caption, we take advantage of GloVe and BERT to compute the semantic disconnection in a sentence as measurement.

$$std = \frac{1}{n} \|X - \bar{X}\|^{1/2} = \frac{1}{n} \sum (x_i - \bar{x})^2$$

x_i represents the value of each dimension of the word-embedding vector

To generate captions with high incongruity, we adjust the original cross-entropy loss function to the following function:

$$L(\hat{y}, y) = H(\hat{y}, y) - \lambda_1 std$$

$H(\hat{y}, y)$ represents the cross entropy of \hat{y} and y

Secondly, we use the following function to combine cross-entropy and ambiguity to compute loss function for each caption:

$$L(\hat{y}, y) = H(\hat{y}, y) - \lambda_2 \prod_i^n num_of_meanings(x_i)$$

We incorporate cross training in the task. Each 200 epochs the loss function change once. Unfortunately, time is limited for this part of work and results for this is not provided. However, the code for this is available. Besides, it is the first try to introduce latent semantic features to image captioning field. So more experiments are absolutely needed.[10] We will continue researching on this.

4 Results

It is a pity that we cannot demonstrate our final output and result due to the limited time. It really took us a long time to understand the mechanisms of the whole model and to search for improvements and modifications. What's even worse, we have encountered some unexpected difficulties during our experiments, which also costs a long time for us to adjust the datasets and debug our program. More Works will be done get the final outputs and illustrate the final results in the future. Some intermediate results are shown below.

Results:



Figure 3: Some intermediate results of the neural network, where the left picture is the input and the right one is the output.

5 Conclusion

In this paper, we have explored how to apply deep learning to generate humorous memes given to an image. We pay special attention to modify the encoder-decoder model based on Show and Tell model introduced in 2015. In addition, some works have been done to implement the attention-based LSTM which is suggested to have a significant effect on improving the performance of image captioning. Moreover, we have tried to design a loss function on our own.

We cannot deny that this project is actually a “semi-finished product”. But we gained a lot from this imperfect project. As beginners, it is not easy for us to understand the function of each part of the model and how it works. Even though we finally have a general understanding of a module, we found it still difficult when we implement that in the code. Only when we have a thorough understanding of the module can we truly understand the code and make our adjustments. And that is exactly what we do in the later period of our project. One thing is greatly improved during this experiment — our ability of integrate information and quick study. That is the most important and valuable thing of this assignment.

Reference:

- [1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator, 2015 *IEEE Conference on computer vision and pattern Recognition (CVPR)*, 2015
- [2] Wang, Cheng and Yang, Haojin and Bartz, Christian and Meinel, Christoph, Image Captioning with Deep Bidirectional LSTMs, <http://arxiv.org/abs/1604.00790> ,apr,2016
- [3] Xu, Kelvin and Ba, Jimmy and Kiros, Ryan and Cho, Kyunghyun and Courville, Aaron and Salakhut-dinov, Ruslan and Zemel, Richard and Bengio, Yoshua, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, <http://arxiv.org/abs/1502.03044> , feb,2015
- [4] Gan, C. and Gan, Z. and He, X. and Gao, J. and Deng, L., StyleNet: Generating Attractive Visual Captions with Styles , jul, 2017
- [5] Luong, M.-T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. Conference on Empirical Methods in Natural Language Processing (2015).
- [6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. arXiv preprint, 1512.00567, 2015.
- [7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator,

2015 *IEEE Conference on computer vision and pattern Recognition (CVPR)*, 2015

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, <https://arxiv.org/abs/1512.03385> , dec, 2015

[9] Grammar as a foreign language Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. and Hinton, G., 2015. Advances in Neural Information Processing Systems, pp. 2773—2781.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[11] D. Yang, A. Lavie, C.Dyer, 2015, Humor Recognition and Humor Anchor Extraction, EMNLP

附录

项目说明

组员：张翠宁、马小涵，卢晴，李睿能

数据来源：

<https://memegenerator.net/>，爬取方式通过 `scraper.py` 给出。

<http://msvocds.blob.core.windows.net/coco2014>，爬取方式通过 `data_download.py` 给出。

参考代码：

本项目想法来自 Abel L. Peirson V 和 E. Meltem Tolunay 两位 Stanford University 的学生，但其项目代码仅公开其基础框架而未公开其细节调整部分，除数据抓取部分外，全部来自 Show and Tell Image Caption Generator (2015) 模型，故所附参考代码链接为原始模型代码链接。

参考论文：<https://arxiv.org/abs/1411.4555>

参考代码：<https://github.com/tensorflow/models/tree/master/research/im2txt>

小组分工：

分工已于代码文件中给出，由于过于分散的缘故，在此集中说明：

数据集构建：张翠宁

词嵌入模型、Loss、TFRecord 文件的构建：李睿能

Attention Mechanism 的加入：卢晴、马小涵

ResNet 替换原 AlexNet：卢晴、马小涵

论文撰写：张翠宁、马小涵，卢晴，李睿能

论文分工：张翠宁 introduction、dataset

马小涵 image caption、CNN、排版、画图

卢晴 画图、decoder、encoder、result、conclusion

李睿能：loss function、word embedding

文件说明：

本项目共含三个子文件夹，一是用于 word-embedding 的 bert fine-tuning 模型文件，二是用于数据抓取的代码文件，三是本项目核心模型文件。

由于本项目所使用的数据集、预训练文件及中间文件过大，时间有限，故只上传代码文件，其中未上传的数据文件有：

数据集部分：Caption.txt, CaptionClean.txt, meme.zip 共近四十万条表情包有关数据，及其所生成的 embedding_matrix.txt, TFRecord 等文件

词嵌入部分：bert_model.ckpt, glove_6B_300d.txt 等文件

AlexNet 预训练权重：bvlc_alexnet.npy 文件