基于 LSTM 网络的中文图片描述

陈斯杰, 韩彪, 马天行

摘　要　本文使用本土中文使用者人工标注的数据集进行了中文图片描述的研究工作。基于 LSTM 网络在 Keras 平台上构建了分词与单字两种中文图片描述模型，经测试二者均能概括出图片中的要点，测试结果的指标值相对令人满意。比较发现，分词模型生成句子似乎更加通顺，而单字模型生成的句子中存在错别字现象，这一现象的内在原因有待进一步研究。

关键词　中文图片描述，LSTM，Keras

1　引　言

1.1 图片描述

随着机器翻译和大数据的兴起，出现了图片描述（Image Caption）的研究浪潮。图片描述是一个融合计算机视觉、自然语言处理和机器学习的综合问题，其主要任务是使机器自动地用自然语言描述给定图片中的内容。显然这是一项很有价值的工作，比如可以帮助视力受损的人们来了解网络图片中的内容[1]。当然这也是一项具有挑战性的工作，可以说这项工作要比图片分类或者目标检测难度更大[2]，因为它不仅要求捕捉到图片中的各个目标，还要能够表达出目标之间的相互关系、目标本身的属性以及它们所进行的活动，而且还要求这些语义知识必须通过某种自然语言来表达[1]。

1.2 中文图片描述

对给定图片进行英文描述的研究在学术界已不少见[1,3,4]，而对采用其他语言进行图片描述的研究工作尚且为数不多。其他语言的图片描述不是简单地在现有模型上改变一下数据集的重复性的工作，而是一个颇具科学研究价值的课题，这主要体现在如下两个方面[5]：

（1）对于图片描述，我们自然希望机器能够抓住图片中的重点内容进行准确的表述，然而在不同的语言文化背景下，人们对同一张图片中何为“重点”本就有很不相同的认知。因此这一研究可有利于探索不同文化背景下的人们如何描述其视觉世界，进而为不同的语言之间提供互相补充的信息。

（2）英语和汉语哪个对人类而言更难学？这一问题至今尚存争议。通过研究计算机采用这两种语言进行图片描述的行为，对理解上述问题将会很有启发。

本文的主要工作正是使用中文标注的图片作为训练集建立模型，实现图片的中文描述。

2　方　法

2.1　数据采集

数据采集在模式识别系统的设计循环中占有相当大的比重。虽然可以采用较小的典型样本集进行初步的可行性研究，但为了确保实际工作时的良好性能，必须要采集到足够多的样本数据[6]。目前做图片描述研究经常用到的经典的数据集有 Flickr8k、Flickr30k、MSCOCO 等[7]，但是这些数据集均是采用英文标注的数据集，无法直接用于中文图像描述任务。一种便捷且廉价的做法是通过机器翻译将英文描述翻译为相应的中文描述后再使用[8]，但文献[5]举例论证了该处理方法的不可靠。诚然，本土的汉语言使用者对图片做的人工标注能够反映语言最真实的使用习惯，无疑对本研究更有价值。本文所采用的数据集即通过本土汉语言使用者人工标注获得。

2.2　特征提取

图片特征采用 VGG19 网络提取。此处的 VGG19 网络是用 ImageNet 数据集预训练的，其输出向量可以作为图像全局特征的一种压缩表示。

2.3 核心模型

2.3.1 基本原理

本文采用的 RNN 模型能够实现输入一个图片的特征向量并输出相应的汉语句子描述。在本模型中，每一个句子被处理成一系列基本元素所构成的序列，这里的“元素”可以是汉语词语或者单个汉字。模型在产生句子时，根据句子当前的元素和前文已经生成的元素，以及图片的特征信息，通过预测下一个元素的条件概率分布，选取概率最大的那个元素作为下一个字或词，从而生成整句话。在第一步，模型仅仅根据图片的特征向量来预测第一个元素的概率分布。设 $\boldsymbol{\theta}$ 为模型参数，则生成一句话的概率为 $p(\boldsymbol{S}\mid\boldsymbol{I};\boldsymbol{\theta})$，其中 $\boldsymbol{I}$ 为图片的特征向量，$\boldsymbol{S}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots\boldsymbol{x}_T)$ 一系列通过 one-hot 编码的词向量的组合，也即一个句子。于是使用链式法则易得对数概率的表达式如下

$$\log p(\boldsymbol{S}\mid\boldsymbol{I};\boldsymbol{\theta})=\sum_{t=0}^{T+1}\log p(\boldsymbol{x}_t\mid\boldsymbol{I},\boldsymbol{x}_0,\cdots\boldsymbol{x}_{t-1};\boldsymbol{\theta}) \quad (1)$$

其中 $\boldsymbol{x}_0=<\mathrm{START}>$，$\boldsymbol{x}_{T+1}=<\mathrm{END}>$ 这两个特殊的元素分别表示一个句子的开始和终止，这样处理可使得模型能够生成不同长度的句子。

2.3.2 LSTM 网络[9]

式(1)中的条件概率是以迭代的方式通过 LSTM 网络估计得出的。LSTM 块如图 1 所示，它除了外部的 RNN 循环外，还具有内部的“LSTM 细胞”循环（自环），因此 LSTM 网络不是简单地

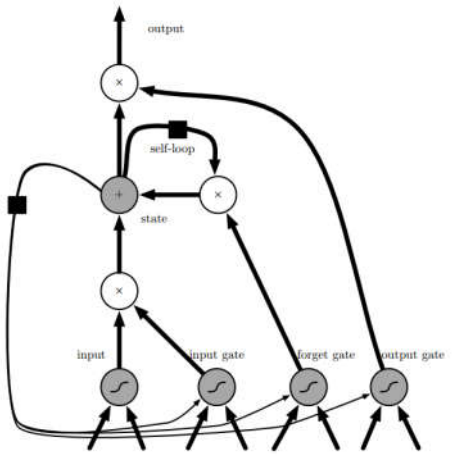

图 1 LSTM 循环网络“细胞”的框图

向输入和循环单元的仿射变换之后施加一个逐元素的非线性。与普通的循环网络类似，每个单元有相同的输入和输出，但也有更多的参数和控制信息流动的门控单元系统。最重要的组成部分是状态单元 $s_i^{(t)}$。其自环的权重（或相关联的时间常数）由遗忘门 $f_i^{(t)}$ 控制（时刻 t 和细胞 i），由 sigmoid 单元将权重设置为 0 和 1 之间的值：

$$f_i^{(t)}=\sigma\left(b_i^f+\sum_j U_{i,j}^f x_j^{(t)}+\sum_j W_{i,j}^f x_j^{(t\text{-}1)}\right) \quad (2)$$

其中 $\boldsymbol{x}^{(t)}$ 是当前的输入向量，$\boldsymbol{h}^{(t)}$ 是当前的隐藏层向量，$\boldsymbol{h}^{(t)}$ 包含所有 LSTM 细胞的输出。$\boldsymbol{b}^f$、$\boldsymbol{U}^f$、$\boldsymbol{W}^f$ 分别是偏置、输入权重和遗忘门的循环权重。因此 LSTM 细胞内部状态以如下方式更新，其中有一个条件的自环权重 $f_i^{(t)}$：

$$s_i^{(t)}=f_i^{(t)}s_i^{(t-1)}+g_i^{(t)}\sigma\left(b_i+\sum_j U_{i,j}x_j^{(t)}+\sum_j W_{i,j}h_j^{(t-1)}\right) \quad (3)$$

其中 $\boldsymbol{b}$、$\boldsymbol{U}$、$\boldsymbol{W}$ 分别是 LSTM 细胞中的偏置、输入权重和遗忘门的循环权重。外部输入门单元 $g_i^{(t)}$ 以类似遗忘门（使用 sigmoid 获得一个 0 和 1 之间的值）的方式更新，但有自身的参数：

$$g_i^{(t)}=\sigma\left(b_i^g+\sum_j U_{i,j}^g x_j^{(t)}+\sum_j W_{i,j}^g x_j^{(t\text{-}1)}\right) \quad (4)$$

LSTM 细胞的输出 $h_i^{(t)}$ 也可以由输出门 $q_i^{(t)}$ 关闭（使用 sigmoid 单元作为门控）：

$$h_i^{(t)}=\tanh(s_i^{(t)})q_i^{(t)} \quad (5)$$

$$q_i^{(t)}=\sigma\left(b_i^o+\sum_j U_{i,j}^o x_j^{(t)}+\sum_j W_{i,j}^o x_j^{(t\text{-}1)}\right) \quad (6)$$

其中 $\boldsymbol{b}^o$、$\boldsymbol{U}^o$、$\boldsymbol{W}^o$ 分别是偏置、输入权重和遗忘门的循环权重。在这些变体中，可以选择使用细胞状态 $s_i^{(t)}$ 作为额外的输入（及其权重），输入到第 i 个单元的三个门，如图 1 所示。这将需要三个额外的参数。

2.3.3 模型实现

采用 LSTM 网络，本模型的迭代过程可表示如下：

$$\boldsymbol{x}_{-1}:=\boldsymbol{W}_e\cdot CNN(\boldsymbol{I}) \quad (7)$$

$$p_0,\boldsymbol{s}_0,\boldsymbol{h}_0\leftarrow LSTM([\boldsymbol{x}_0;\boldsymbol{x}_{-1}],\boldsymbol{0},\boldsymbol{0}) \quad (8)$$

$$p_{t+1},\boldsymbol{s}_{t+1},\boldsymbol{h}_{t+1}\leftarrow LSTM([\boldsymbol{x}_t;\boldsymbol{x}_{-1}],\boldsymbol{s}_t,\boldsymbol{h}_t) \quad (9)$$

整个模型的实现流程如下图所示：

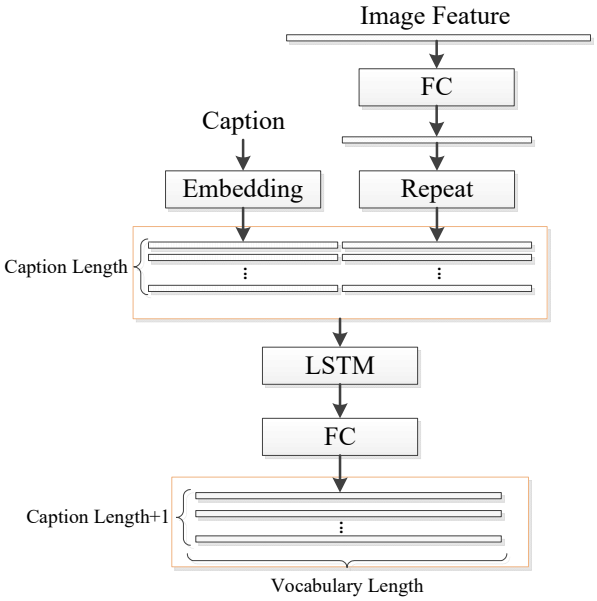

图 2　模型示意图

2.4　模型训练

训练本模型的过程就是在给定图片特征向量 $\boldsymbol{I}$ 和句子元素 $\boldsymbol{x}_t$ 及其前文信息（隐含于 $\boldsymbol{h}_{t-1}$ 中），使模型预测句子中下一个元素的概率分布 p_t 的过程。如式(7)~(9)所示，初始情况下置 $\boldsymbol{h}_0=\boldsymbol{0}$，$\boldsymbol{x}_0=<\mathrm{START}>$，根据图片特征 $\boldsymbol{x}_{-1}$ 来估计句子下一个元素 $\boldsymbol{x}_1$ 的概率分布，以图片实际标注的句子中的第一个词作为标签来确定概率值的表达。然后再根据句子第一个元素 $\boldsymbol{x}_1$ 以及 $\boldsymbol{h}_1$ 来预测第二个元素 $\boldsymbol{x}_2$ 的概率分布，以此类推，最终根据句子最后一个元素 $\boldsymbol{x}_T$ 来预测下一个特殊的元素 $<\mathrm{END}>$，以此作为终结，从而得到式(1)给出的关于参数 $\boldsymbol{\theta}$ 的对数似然函数。通过极大化该似然函数来实现参数的训练。

2.5　模型测试与评估

用训练好的模型来产生一个图片描述句子时，根据图片的特征向量 $\boldsymbol{x}_{-1}$、初始隐藏层 $\boldsymbol{h}_0=\boldsymbol{0}$ 以及句子起始符 $\boldsymbol{x}_0=<\mathrm{START}>$ 来计算句子第一个词语的概率分布，从该分布中选取最大概率所对应的词语作为句子的第一个词 $\boldsymbol{x}_1$，然后再由 $\boldsymbol{x}_1$ 预测 $\boldsymbol{x}_2$，以此类推，直到预测得到句子终止符 $<\mathrm{END}>$ 为止，这样就得到了一个完整的句子。最终通过生成句子的 BLEU、ROUGE、CIDEr 等指标来评估模型预报结果的好坏。

3　结　果

3.1　数据集概况

本数据集是由本土汉语言使用者人工标注得到，分为训练集、验证集和测试集。其中训练集有 8000 张图片，验证集和测试集分别有有 1000 张图片，数据集中每张图片有最多 5 句中文标注。

数据集中所有图片的 CNN 特征均已实现提取完毕，这些特征包括：

A. VGG19 网络的第一个全连接层 fc1 特征，4096 维；
B. VGG19 网络的第二个全连接层 fc2 特征，4096 维；
C. VGG19 网络全连接层之前的卷积特征，7*7*512 维。

其中 A 和 B 是图像的整体特征，C 中包含图像的位置特征，每一个长度为 512 的向量对应于将原图划分为 7*7 的区域后，以每个区域为中心得到的征。

这里，我们仅选用图片的整体特征 fc2 进行实验。

3.2　模型参数

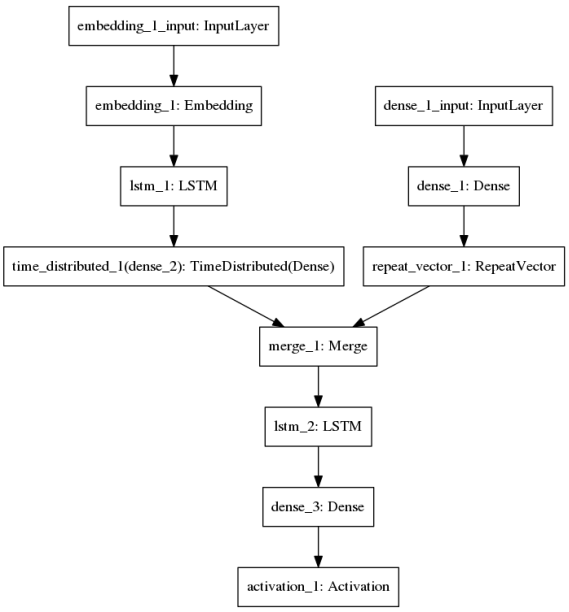

图 3　模型参数示意图

本文构建了语言子模型和图像子模型：

语言子模型负责将分割后的中文文本单元（单个汉字或者单个词语）嵌入到一个较低维的向量中去，从而起到压缩空间的作用。在分词模型中句子

最长长度为 30，在单个汉字模型中句子最长长度为 50，对应的 Embedding 层输入维度为 30 或 50。LSTM 层的输入和输出皆为 30x256（或 50x256）的矩阵。此处选择 128 维作为嵌入后空间的维度（经验选择），故 TimeDistributed 层输出为 30x128 的矩阵。

图像子模型中，输入向量为 4096 维（图像 fc2 特征向量的维度）的向量，经过一个全连接层，输出成为一个 128 维的向量，并经过 Repeat 层叠加扩增与语言子模型的输出合并。合并后的输出经 LSTM 分类预测出一个 One-Hot 的向量，来表示当前图像和上文条件下，预测出的下文第一个词语（或单个汉字）。

在分词模型中，训练采用的 Batch 大小为 1400；在单个汉字模型中，训练采用的 Batch 大小为 960。

3.4 运行平台

本文使用了清华信息科学与技术国家实验室生物信息学研究部提供的深度学习工作站，CPU 为 AMD Ryzen7 1700X 8 核心 3.4GHz，DDR4 32GB 内存，使用两块开启 SLI 交火的 NVIDIA GTX 1080Ti 显卡加速神经网络训练。

在此硬件平台之上，本文选用了 CentOS 7.2 操作系统，并在 Anaconda Python 3.5.3 计算平台上部署了 Tensorflow 1.1.0 和 Keras 2.0.2 两个深度学习库，且以 Tensorflow 作为 Keras 的后端。显卡驱动为 NVIDIA 381.22，cuDNN 版本为 5.1，CUDA Toolkit 版本为 8.0。

3.5 优化方法

本文使用了 Keras 框架提供的默认的 RMSProp 优化方法。该方法是一种改进的随机梯度下降方法，并且能够结合前一回合的学习率，进行自动的学习率的调整。

3.6 定量指标

本文采用分词（以词语作为句子基本元素）与不分词（以单个汉字作为句子基本元素）两种方法分别建立了中文图片描述模型，两个模型经训练若干 epoch 后在测试集上的最佳预报结果的各个指标情况如下表所示：

表 1 分词模型与单字模型的测试指标对比

	bleu-1	bleu-2	bleu-3	bleu-4	rouge	cider
分词	0.6200	0.4850	0.3710	0.2790	0.4700	0.9430
单字	0.5980	0.4620	0.3460	0.2590	0.4650	0.8630

从表中可见，采用当前训练集进行训练，分词模型的测试效果要优于单字模型。

模型的训练过程中过拟合现象较为显著，以分词模型为例，在训练过程中，其损失函数值逐渐随着训练 epoch 数的增加逐渐下降（图 4），句子词语预测精度逐渐逐渐上升（图 5）。而随着训练 epoch 数的增加，模型在测试集上预报结果的各项指标却呈现显著的下降，如表 2 所示。

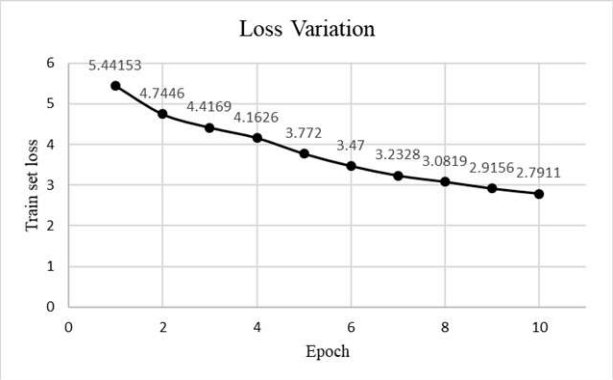

图 4 损失函数值随训练 epoch 数的变化情况

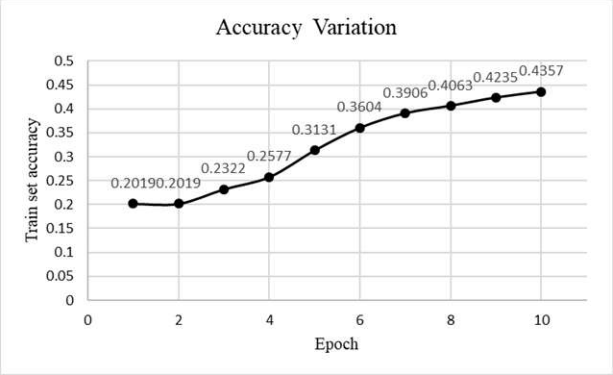

图 5 训练准确度随训练 epoch 数的变化情况

表 2 分词模型训练不同 epoch 的测试指标

	bleu-1	bleu-2	bleu-3	bleu-4	rouge	cider
9	0.6200	0.4850	0.3710	0.2790	0.4700	0.9430
20	0.6060	0.4600	0.3470	0.2610	0.4600	0.8730
25	0.5860	0.4380	0.3210	0.2350	0.4360	0.8050
50	0.5710	0.4220	0.3080	0.2240	0.4330	0.7970

3.7 定性可视化结果

这里给出两种模型（分词和单字）在测试集上生成的句子的几个示例。

- 例 1：

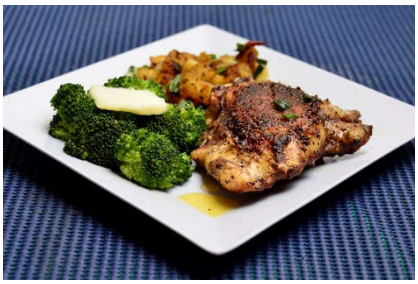

图 6 测试集样本 9543 原图

分词模型预报结果：

- 盘子里放着一些食物。（9 epoch）
- 餐盘里有西兰花和肉。（20 epoch）
- 白的盘上有牛排和肉。（25 epoch）

单字模型预报结果：

- 盘子里有一个盘子里有一些食物。（5 epoch）
- 白色的盘子里有一些食物。（36 epoch）
- 一盘牛排沙拉的食物（50 epoch）

- 例 2：

图 7 测试集样本 9503 原图

分词模型预报结果：

- 草地上趴着一只小狗。（9 epoch）
- 一只狗站在草地上。（20 epoch）
- 草地上有一只山羊。（25 epoch）

单字模型预报结果：

- 草地上有一只狗在草地上。（5 epoch）
- 一只白色的小狗站在草地上。（36 epoch）
- 草地上有一只红白项圈的小消防栓。（50 epoch）

- 例 3：

图 8 测试集样本 9823 原图

分词模型预报结果：

- 一个男人在街上行走。（9 epoch）
- 一个女人走在街上走。（20 epoch）
- 三个人在广告牌粉红色的大街上。（25 epoch）

单字模型预报结果：

- 一个男人正在路上。（5 epoch）
- 一个穿着蓝色衣服的男人打着一个黑色的行人。（36 epoch）
- 三个人在两名的街道上休息。（50 epoch）

分析上面的几个示例不难得到以下两点结论：

1. 训练的 epoch 数越多，生成的句子越倾向于对图片中的细节进行描述。（结合表 2 中的结果分析可见，BLEU、ROUGE、CIDEr 等指标对生成句子的要求有过于谨慎之嫌。）
2. 分词模型和单字模型均能概括出图片中的一些要点，但二者相比，分词模型生成的句子似乎更通顺一些。

此外在其他测试样本中还存在一个值得注意的现象，单字模型生成的句子中相对容易出现错别字（我们知道，对人工智能而言，这样的错误通常是难能可贵的），比如以下例子均是把“在”错误地用成了“再”：

- 样本 9018：一个穿着黑色衣服的女人**再**行驶。

- 样本 9045：一个穿黑色衣服的男孩**再**玩滑板。
- 样本 9681：一个光头男子**再**吃面包。

4 讨 论

本文使用本土中文使用者人工标注的数据集进行了中文图片描述的研究工作。基于 LSTM 网络在 Keras 平台上构建了分词与单字两种中文图片描述模型，通过比较二者在测试集上生成句子的指标可发现，在当前数据集下，分词模型描述结果的各项指标要优于单字模型。就模型的训练过程而言，过拟合现象较为显著，对于分词模型，训练 9 个 epoch 后在测试集上的指标就呈现出下降趋势。而这些指标也仅能作为衡量模型生成句子好坏的一种参考，从模型实际生成的句子来看，虽然经过长时间训练所生成的句子的指标不高，但这些句子似乎更能命中图片中的细节。直观上来看，分词模型与单字模型均能概括出图片中的一些要点，但进一步比较二者发现，分词模型生成句子相对更加通顺。另外值得注意的是，单字模型生成的句子中存在错别字现象，这一现象的内在原因有待进一步研究。

致 谢 感谢清华大学自动化系 BigEye 实验室组织收集了文中使用的中文图片描述数据集；感谢清华大学自动化系生物信息学研究部提供计算资源。

参 考 文 献

[1] Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3156-3164.

[2] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[3] Fang H, Gupta S, Iandola F, et al. From captions to visual concepts and back[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1473-1482.

[4] Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3128-3137.

[5] Li X, Lan W, Dong J, et al. Adding Chinese captions to images[C]//Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval. ACM, 2016: 271-275.

[6] Duda R O, 杜达, Hart P E, et al. 模式分类[M]. 机械工业出版社, 2003.

[7] PaperWeekly 第二十二期——Image Caption 任务综述：http://mp.weixin.qq.com/s?__biz=MzIwMTc4ODE0Mw==&mid=2247484014&idx=1&sn=4a053986f5dc8abb45097fed169465fa&chksm=96e9ddeea19e54f83b717d63029a12715c238de8d6af261fa64af2d9b949480e685b8c283dda&scene=21#wechat_redirect

[8] Peng H, Li N. Generating Chinese Captions for Flickr30K Images[J].

[9] Bengio Y, Goodfellow I J, Courville A. Deep learning[J]. Nature, 2015, 521: 436-444.