基于 LSTM 的图像中文描述

窦　珊　　2016211096,　　ds16@mails.tsinghua.edu.cn
马平烁　　2016211097,　　mps16@mails.tsinghua.edu.cn
汪　洁　　2016211151, wangjie16@mails.tsinghua.edu.cn
* 清华大学自动化系，北京 100084

摘要：自动描述图像内容(Image Caption)是计算机视觉和自然语言处理的一个交叉问题，本文使用卷积神经网络 (Convolutional Neural Network, CNN) 模型抽象图像，并用 LSTM (Long Short Term Memory) 模型生成流畅的中文描述。在此基础上，本文将 Attention 机制加入 LSTM 模型，作为第二种模型进行对比。最终我们在 9000 张图片的中文标注数据集上进行训练，证实了基于 LSTM 的图像中文描述模型的可行性和优越性。

1 引言

自动描述图像内容是一项极具挑战性的工作，图像和自然语言这两种数据的表达方式完全不同：图像的描述一般是二维的矩阵，而自然语言的描述一般是一个序列。更细致地来讲，东方国家的象形文字和西方国家的拼音文字，表达方式以及语法也有很大不同。如何选用合适的模型来抽象和表达这些数据一直是近些年来的工作重点。

我们的工作目标是解决图像的中文自动描述。Oriol Vinyals[1]等人在 2015 年已经研究过图像的英文自动描述，他们使用卷积神经网络[2] (Convolutional Neural Network, CNN) 来抽象图像，使用递归神经网络[3] (Recurrent Neural Network, RNN) 将抽象过的图像翻译为英文。其中的 CNN 已经被多次证实可以充分、贴切地抽象一副图像；而 RNN 则是广泛应用于机器翻译、自动对话领域的自然语言生成模型，它可以描述句子中词语的顺序关系，就像人去读一个句子一样。因此我们决定采用这一模型，将其应用在中文描述上。该模型采用最大似然的思想，对输入的图像 I，训练使其似然估计$p(S|I)$最大，其中S是生成的中文描述。

人类视觉系统很有趣的一个特点就是注意力(Attention)的存在，当我们在观察一张照片的时候，我们往往动态的关注想要获取的图片特征信息，而不是将整张照片压缩为静态表示，当我们面临许多图片时，这个特点显得尤为重要。Bahdanau[4]等人在 2014 年成功的将 Attention 机制应用到机器翻译中，被证实可

以高准确度的进行句子翻译，受此启发 Kelvin[5] 等人在 2015 年将 Attention 机制运用到图像英文描述中，在常见的 LSTM 结构下，引入两种不同的 Attention 机制图像描述模型，其一："soft"是通过标准反向传播算法训练确定性 Attention 机制；其二："hard"是通过最大化似然函数下界训练的随机性 Attention 机制。在这两种机制任一种的作用下，每一次迭代中可以选择关注图片中不同位置的不同特征，将其应用到 LSTM 结构中进行图片描述。基于 Attention 的图片描述方法已经被证明在大量图片的英文描述中有最高水平的表现，因此我们决定采用这一模型，将其使用到中文描述上，作为一种改进的中文图片描述方法。

基于图片特征和中文描述，我们实现的主要工作如下：1）使用已有的图片特征(1*1*4096)，加上对应的中文描述，使用随机梯度下降，训练出 LSTM 序列模型；2）使用 BeamSearch 的方法，利用训练好的 LSTM 模型对测试集的图片生成中文描述；3）使用图片的卷积特征（7*7*512），利用 Attention 的机制，训练出 LSTM 序列模型并生成描述。我们使用清华大学大眼睛实验室[1]收集到的 9000 张图片的中文标注数据集进行训练，取得了很好的效果，其中基于 Attention 机制的 LSTM 在 BLEU-1 上最高得分 0.681，CIDEr 上最高得分 1.162。

2 模型和方法

我们总共实现了两种模型：1）基础模型：使用 LSTM 生成图像描述；2）改进模型：使用 Attention 机制生成图像描述。后者是对前者的改进，下面分别介绍两个模型的实现方法。

2.1 基础模型：使用 LSTM 生成图像描述

我们的基础模型是端到端的模型，简单来说，我们的目标是直接最大化生成正确中文描述的概率，即：

$$\theta^* = arg\max_{\theta} \sum_{(I,S)} logp(S|I;\theta) \tag{1}$$

其中θ是模型的参数，I 是输入的图像，S 是正确的中文描述。S 可以代表各种不同的句子，其长度是不固定的。我们假设一个句子 S 是由$S_0, S_1, \ldots, S_{N-1}$构成，

[1] 清华大学大眼睛实验室: http://bigeye.au.tsinghua.edu.cn

那么对一个给定的句子 I，产生描述 S 的概率可以表述为：

$$logp(S|I) = \sum_{t=0}^{N} logp(S_t|I, S_0, \dots S_{N-1}) \quad (2)$$

也即是说，的模型可以由一个一个的小模型组合而成，每个模型的输入都是一张图像和已有序列，输出是产生下一个字的概率。通常这个模型可以用 RNN 来描述，而 LSTM[6]是最常用的一种 RNN 模型。

我们的模型主要分为三个部分。第一部分是 CNN(Convolutional Neural Network)，它将 2 维图像抽象为 1 维的向量；第二部分是 LSTM(Long-Short Term Memory)，是将图像含义“翻译”成中文的核心模块；第三部分是 Word Embedding 部分，无论是汉字还是英文单词，通常都是由一种称作 OneHot 的方式描述，为了更好地对词义表达，通常会先进行 Word Embedding。下面分别对 Word Embedding 模块和 LSTM 生成语句模块进行介绍。

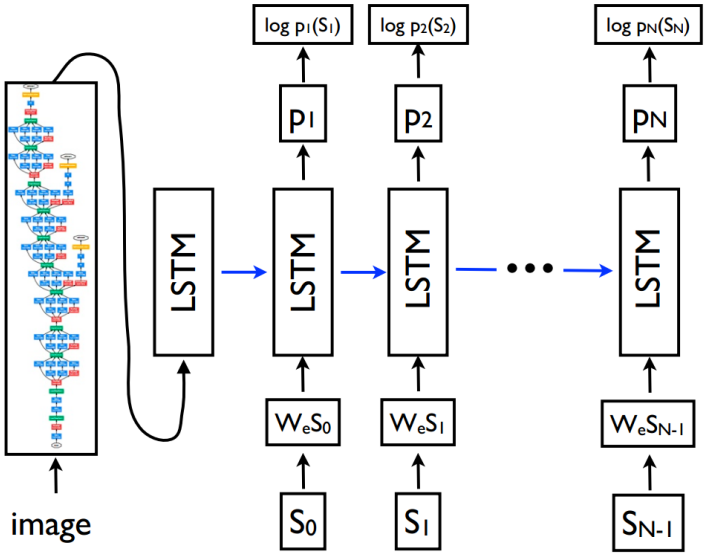

图 2.1 图像自动中文描述模型[1]

2.1.1 LSTM 生成语句模块

在图像的中文描述中，LSTM 担任了文本生成的角色。图 2.1 展示了 LSTM 模块之间联系的流程。整个语言生成部分由 N 个 LSTM 构成，这里的 N 指的是

[1] 引用自 Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator

我们生成的句子长度。

图中蓝色的线是 LSTM 中的隐藏层，不妨粗略地称为h_t，对于第 t 个 LSTM 子模块，有：

$$h_{t+1} = f(h_t, x_t) \tag{3}$$

其中x_t为该 LSTM 的输入。同时该模块会输出p_t，表示输出每个词语的概率，p_t是$1 \times v$维的向量，其中 v 表示字典的长度。

第一个 LSTM 的输入（即x_0）是抽象过的图像特征，之后第 t 个 LSTM 的输入x_{t-1}是中文描述中的第 t-1 个字/词。这样，对模型输入一幅图像，即可以生成一个长度为 N 的句子。

事实上，LSTM 的隐藏层并不是一个简单的一维的向量h_t，而是由很多复杂的门来控制。LSTM 模型的核心是细胞状态 c，直接在整个链上运行，只有一些少量的线性交互，如图 2.2，信息可以在上面流传、保持不变。

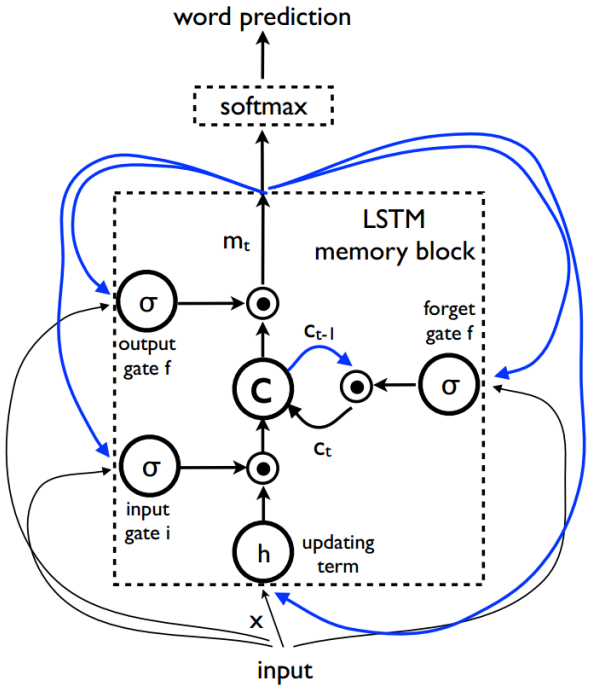

图 2.2 LSTM 模型细节[1]

LSTM 通过精心设计的称作为“门”的结构来去除或者增加信息到细胞状态的能力。门是一种让信息选择式通过的方法。一般来说，LSTM 会维护三个门，控

[1] 引用自 Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator

制是否将信息去除(forget gate *f*)，是否增加信息(input gate *i*)，是否输出细胞状态(output gate *o*)。各个门和细胞状态之间的更新方式如下：

$$o_t = \sigma(W_o x_t + U_o m_{t-1} + b_o) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f m_{t-1} + b_f) \tag{5}$$

$$i_t = \sigma(W_i x_t + U_i m_{t-1} + b_t) \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx} x_t + W_{cm} m_{t-1}) \tag{7}$$

$$m_t = o_t \odot c_t \tag{8}$$

$$p_{t+1} = Softmax(m_t) \tag{9}$$

其中⊙表示对位相乘，W 表示训练后的参数，$\sigma(\cdot)$表 sigmoid 函数，$h(\cdot)$表示 tanh 函数。最后公式中的m_t表示 softmax 的输入向量。Softmax 的输出p_t即为所有字/词的概率分布。

2.1.2 Word Embedding 过程

我们使用 one-hot 向量S_t来表示字典里的每个中文字或中文词，其中S_0和S_N表示特殊的字符，比如用$和%分别表示句子起始和终止。词向量和图片特征被映射到同一高维空间，前者通过 word embedding 实现，后者通过 CNN 实现。

中文和英文的区别是中文是由“字”构成了，一个或者多个“字”构成一个词。所以在 one-hot 的过程中可以选择按照“字”生成字典，或者按照“词语”生成字典。我们实验中发现分词后的结果要好很多，我们使用清华大学自然语言处理与社会人文计算实验室[1]研发的分词软件包 THULAC 进行中文的分词工作。

2.2 改进模型：使用 Attention 机制生成图像描述

2.2.1 整体结构

基于 Attention 机制的中文图片描述从输入到输出包括编码和解码两个部分。如图 2.3 所示

[1] 清华大学自然语言处理与社会人文计算实验室主页: http://www.thunlp.org/site2/

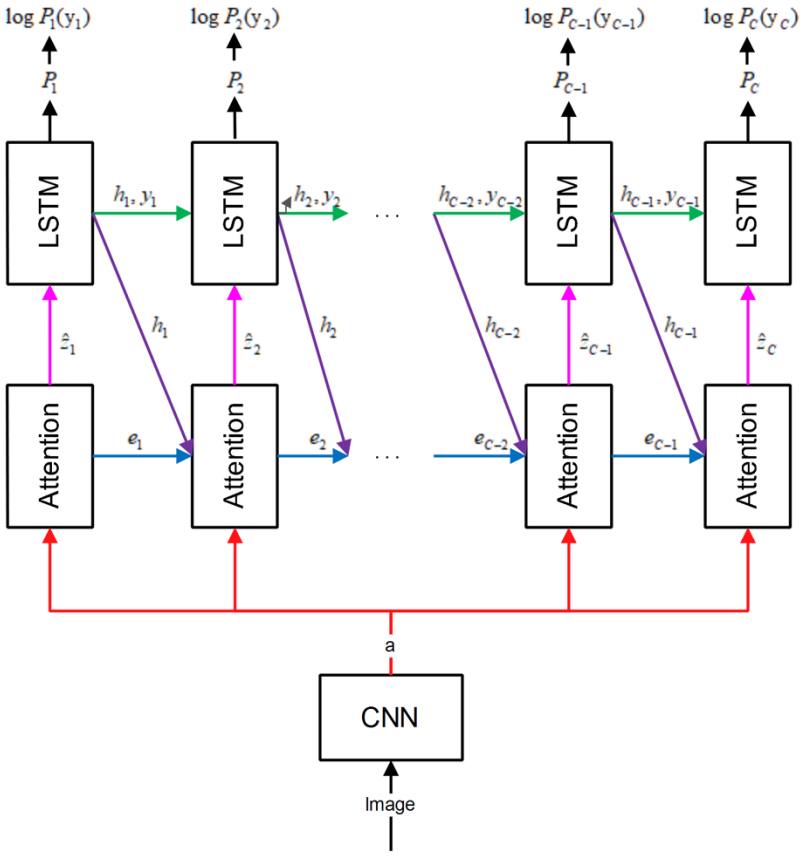

图 2.3 基于 Attention 机制的中文图片描述结构

其中，Image 代表彩色图像，使用 CNN 图像卷积网络提取图像的 L 个特征向量，每个特征向量是一个 D 维特征，用来描述图像的不同位置，表示如下：

$$\mathrm{a} = \{a_1, \cdots, a_i, \cdots, a_L\}\ ,\ \ a_i \in R^D \tag{10}$$

$\hat{z}_t$也是一个 D 维特征，共有 C 个，表示每个单词对应的上下文，不同于a_i是在特征提取中一次生成的，而$\hat{z}_t$是一个逐步生成的时间序列，因此用下标 t 表示。其中，C 代表中文描述的长度，长度 C 不定。

$$\hat{z} = \{\hat{z}_1, \cdots, \hat{z}_t, \cdots, \hat{z}_C\}\ ,\ \ \hat{z}_t \in R^D \tag{11}$$

e_t是 Attention 中隐藏层状态向量,记录之前对图片的关注状态。

$$\mathrm{e} = \{e_1, \cdots, e_t, \cdots, e_C\} \tag{12}$$

y_t是有序的中文描述，共有 C 个，是 K 维向量，K 代表词典长度，表示如下：

$$y = \{y_1, \cdots, y_t, \cdots, y_C\}\ ,\ y_t \in R^K \tag{13}$$

Attention 机制选取 soft 确定性机制，LSTM 为 C 个 LSTM 结构串联组成.

2.2.2 基于 Attention 机制的 LSTM 网络结构

由于编码过程的卷积特征提取未包含在本次工作内，因此以下主要描述解码过程。

解码过程使用 LSTM 网络结构，与基础 LSTM 模型中不同的是，这里的 LSTM 结构除了结合上一隐藏层输出h_{t-1}之外，还利用了上一层中文字输出y_{t-1}和由 Attention 机制产生的上下文描述向量z_t.具体的改进 LSTM 结构如图 2.4 所示。

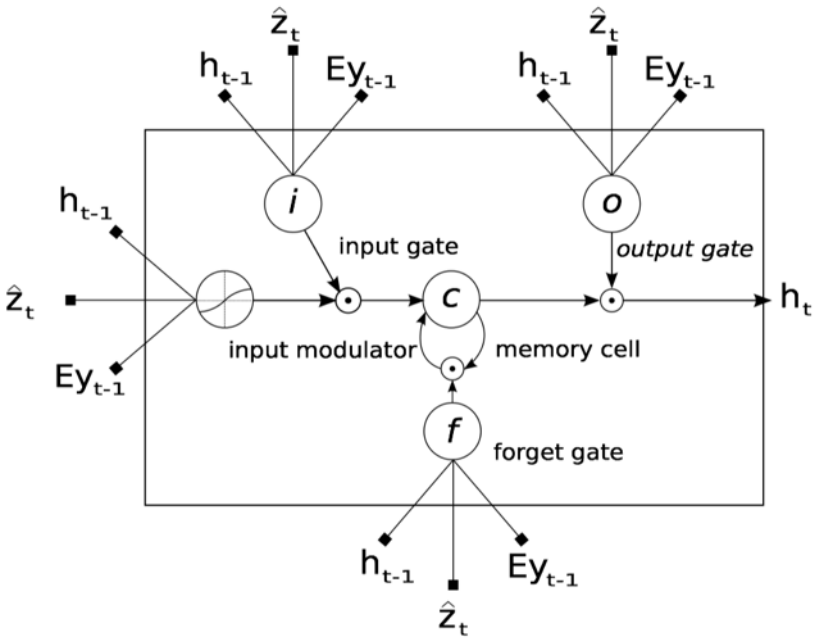

图 2.4 Attentionz 机制中的单个 LSTM 结构[1]

类似的，LSTM 同样会维护三个门，控制是否将信息去除(forget gate *f*)，是否增加信息(input gate *i*)，是否输出细胞状态(output gate *o*)。不同的是三个门和各个胞之间的更新方式如下：

[1] 引用自 Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generatin with visual Attention.

$$i_t = \sigma(W_i E y_{t-1} + U_i h_{t-1} + Z_i \hat{z}_t + b_i) \quad (14)$$

$$f_t = \sigma(W_f E y_{t-1} + U_f h_{t-1} + Z_f \hat{z}_t + b_f) \quad (15)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c E y_{t-1} + U_c h_{t-1} + Z_c \hat{z}_t + b_c) \quad (16)$$

$$o_t = \sigma(W_o E y_{t-1} + U_o h_{t-1} + Z_o \hat{z}_t + b_o) \quad (17)$$

$$h_t = o_t \, tanh(c_t) \quad (18)$$

其中$W_■, U_■, Z_■, b_■$是待训练参数，E $\in R^{m \times K}$是 embedding 矩阵，m 和 n 分别代表 Embedding 和 LSTM 的维数。

通常情况下，特征向量$\hat{z}_t$是 t 时刻我们所关注的图片相应部分的动态表示。定义机制ϕ通过正项权重α_i将图片原始特征$a_i, i = 1, \cdots, L$映射到$\hat{z}_t$，在每个位置i处的正项权重α_i(可以理解为位置i获得的关注)通过 Attention 机制f_{att}结合上一隐藏层输出h_{t-1}计算得到，因为下一步“看哪儿”不但与实际图像有关，还受之前看到的东西的影响。具体计算过程表示如图 2.5 所示。

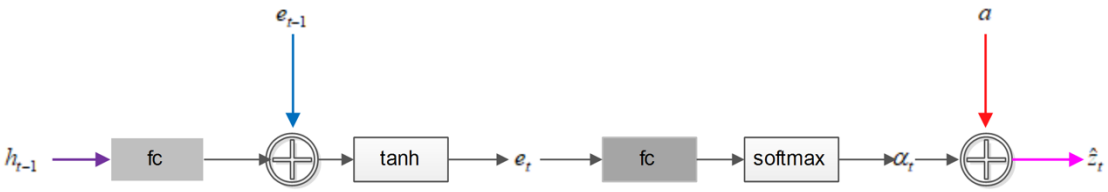

图 2.5 “soft” Attention 机制结构

上图中，阴影部分 fc 代表参数计算，用公式表示如下：

$$\mathrm{e}_{ti} = f_{att}(a_i, h_{t-1}) \quad (19)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})} \quad (20)$$

$$\sum_{\mathrm{i}} \alpha_{ti} = 1 \quad (21)$$

$\hat{z}_t$计算公式为,ϕ的具体细节将在下一部分详细阐述。

$$\hat{z}_t = \phi（\{a_i\}, \{\alpha_i\}） \quad (22)$$

LSTM 结构的初始状态h_0完全由图像原始特征a_i决定，在上述模型的作用下，每个字输出的概率也可以表示为[7]

$$p(y_t|\mathrm{a}, y_1^{t-1}) \propto \exp(L_0(\mathrm{E}y_{t-1} + L_h h_t + L_z \hat{z}_t)) \tag{23}$$

其中$L_0 \in R^{K\times m}, L_h \in R^{m\times n}, L_z \in R^{m\times D}, E$是随机初始化的待训练参数。

2.2.3 确定性“soft” Attention 机制

在上部分中我们提到 Attention 机制f_{att}，本部分将详细讨论本程序中使用到的“soft” Attention 机制。

使用位置变量s_t表示在生成第 t 个字时的模型关注中心，则$s_{t,i}$=1，如果 t 时刻关注中心为 i，否则为 0. 将$s_{t,i}$处理为中间潜变量，则可以令

$$\mathrm{p}(s_{t,i} = 1 | s_{j<t}, a\ ) = \alpha_{t,i} \tag{24}$$

$$\hat{z}_t = \sum_i s_{t,i} a_i \tag{25}$$

取$\hat{z}_t$的期望为

$$E_{\mathrm{p}(s_t|\mathrm{a})}[\hat{z}_t] = \sum_{i=1}^{L} \alpha_{ti}\, a_i \tag{26}$$

则确定性 Attention 机制可以表示为

$$\phi(\{a_i\}, \{\alpha_i\}) = \sum_{i=1}^{L} \alpha_i\, a_i \tag{27}$$

在模型训练的过程中，为了使模型对于图片中的每一部分有相同的关注度，因此在任意时刻设置$\sum_{\mathrm{i}} \alpha_{ti} = 1$，但是这种做法在解码的过程中有可能使图片中一些部分的部分被忽略，为了缓解这种问题，可以令$\sum_{\mathrm{i}} \alpha_{ti} \approx \tau, \tau \geq \mathrm{L/D}$;除此之外，soft Attention 模型还会从前一隐藏层h_{t-1}中预测一个门限系数β，因此

$$\phi(\{a_i\}, \{\alpha_i\}) = \beta \sum_{i=1}^{L} \alpha_i\, a_i \tag{28}$$

$$\beta_t = \sigma(f_\beta(h_{t-1})) \tag{29}$$

β决定在每一步解码过程中更多的关注语言模型还是$\hat{z}_t$。则如果取$\tau = 1$ soft

Attention 模型通过最小化如下的惩罚负对数函数来进行端到端训练。

$$L_d = -\log\big(p(y|a)\big) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2 \tag{30}$$

3 结果

我们使用清华大学大眼睛实验室[1]收集到的 9000 张图片的中文标注数据集进行训练，其中图片的 CNN 特征已经给出。我们进行了大量的实验来评估模型的有效性，并对模型进行优化以提高中文描述的生成效果，包括比较图片的不同特征表示对预测能力的影响，分别使用图像的整体特征和位置特征作为图片的表示。另外，我们对中文描述进行分字和分词两种处理并且分别进行训练。

我们的模型在深度学习框架 TensorFlow 上用 Python3 编程实现，程序运行在一片 NVIDIA Titan X Pascal GPU 上。其中第一个模型的实现参考 Google 的开源代码[2]im2txt，而基于 Attention 的模型实现参考 yunjey 的源码 show-attend-and-tell[3]。

3.1 训练和测试细节

我们的数据集中给出的图像特征是训练好的特征，也即是说我们的 CNN 子模块在我们整个系统中不会更新参数。而 Word Embedding 的参数W_e并不是事先训练好的，而是随着算法迭代不断更新的。LSTM 的内部的隐藏层维度是 512 维，Word Embedding 和图片 Embedding 的维数统一为 4096，作为 LSTM 的输入。使用 Attention 机制时，Attention 输入特征维度为 49×512，是将图片均分为 49 个部分，以每个部分为中心提取得到，LSTM 的输入维度相应调整为 512 维度。

我们使用随机梯度下降法(Stochastic Gradient Descent, SGD)和固定的学习率进行模型优化，模型的初始参数是随机的。为了解决过拟合问题，我们对模型添加了 dropout 机制。

在测试阶段，我们使用了 BeamSearch 机制，也即是说第 t 个 LSTM 维持最大可能的 k 个句子的概率，在 t+1 个 LSTM 中分别输入这 k 个句子，并且依此计

[1] 清华大学大眼睛实验室: http://bigeye.au.tsinghua.edu.cn

[2] Im2txt 源码地址: https://github.com/tensorflow/models/tree/master/im2txt

[3] Show-attend-and-tell 源码地址: https://github.com/yunjey/show-attend-and-tell

算出新的 *k* 个最大概率的句子。最终模型会输出 *k* 个句子，以及它们的概率。实验中发现 *k*=4 的时候效果最好。不过遗憾的是，我们目前只在第一个模型上实现了 BeamSearch 机制，而尚未在 Attention 模型中加入该机制，这是模型进一步优化的突破点之一。

3.2 评价指标

评价生成的语句是否“正确”是很困难的一件事，因为自然语言是非常复杂的。但即使这样，目前已有很多公认比较好的算法指标来评价生成的描述。我们在评估模型时用到了 BLEU-1，BLEU-2，BLEU-3，BLEU-4，ROUGE，CIDEr 几种指标，对它们进行归一化求均值或者一个合理的得分作为评价中文描述的指标。

3.3 定量分析

通过不断调整学习率和 SGD 中的 batch 大小，我们分别在两个模型上得到最佳结果。我们发现在处理中文标注时，分词的结果要比分成单个字的结果好不少，下面的得分也都是基于分词的结果。在表 3.1 中记录了分词后训练的模型，在 1000 张图片的测试集上的最佳测试结果。

表 3.1 两个模型的最佳得分

模型	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	CIDEr
基础模型:LSTM	0.668	0.536	0.426	0.337	0.503	1.116
改进模型:Attention	0.676	0.542	0.431	0.34	0.513	1.162

从表 3.1 来看，Attention 的模型得分要明显高于普通 LSTM 模型要好一些，而且 Attention 模型的收敛时间也要短一些。不过在实验中我们还是观测到了很明显的过拟合现象，我们以 Attention 为例，在图 3.1 中给出它训练到不同 epoch 时的测试得分(对以上 6 个指标得分归一化并且求和)。从图中可以看到 epoch 达到 12 的时候，测试得分已经开始降低了，所以实验中还是存在过拟合的现象，下一步的重点工作之一是减少过拟合问题。

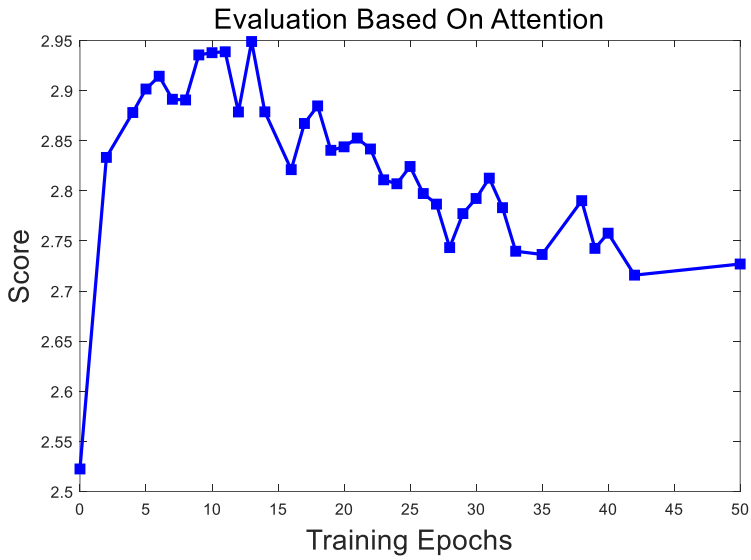

图 3.1　Attention 模型中：随着 epoch 增加的测试得分

图 3.2　LSTM 基础模型生成中文描述可视化。

3.4 定性分析

3.4.1 基础 LSTM 模型的结果分析

LSTM 基础模型生成图片中文描述的过程如图 3.2 所示，可以看到，模型能从训练集学习到图片中相应的人、物和事件；然后在生成的句子中，使用这些出现过的词语，生成新的句子。

3.4.2 基于 Attention 机制的模型结果分析

为了使 Attention 机制中权值α的作用可视化，我们进行权值上采样，对图像添加区域性高斯噪声来模拟权值较重区域的位置。如图 3.3 和图 3.4 所示，其中高亮区域代表关注的位置，彩色的中文描述代表其输出的描述。

图 3.3 关注到图像的正确位置的例子：左图中 Attention 识别了“狗狗”和“飞盘”，右图识别了“鸟”和“石头”。

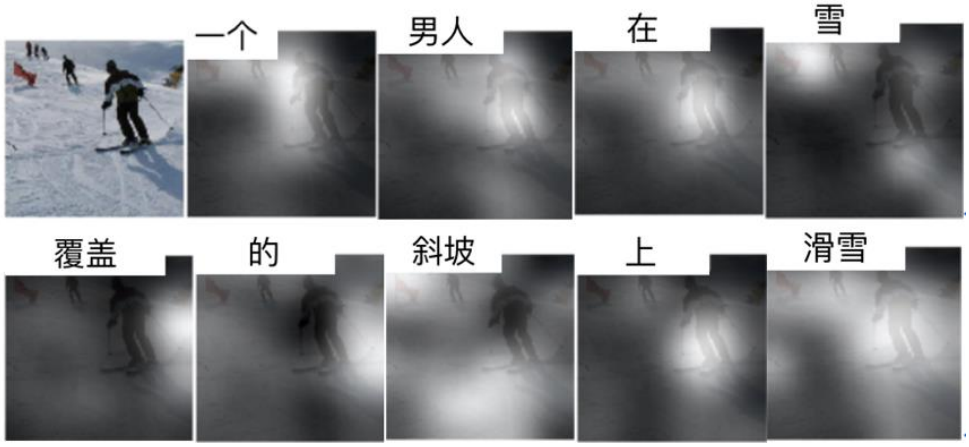

图 3.4 Attention 中每个中文描述对应的图像区域可视化：既关注了“男人”、“斜坡”这些难以辨认的物体信息，也关注到“覆盖”、“上”这些方位特征

在引言中我们提过，人类观察事物的一个有趣的特点就是会选择性的选取事物的部分特征进行观察，而 Attention 模型就是受这一特点的启发而来，在图 3.4 中，图片焦点对准一个男人，背景中还有很多人和山峦，Attention 成功识别了“男

人”、“斜坡”这些关键信息，而远方的“人”和“山峦”则被 Attention 忽略了，这和人类的观察方式是非常相近的。

另外，从可视化结果可以看出，Attention 生成的中文描述与人类语言描述具有相同的连贯性，说明模型在关注某一特征时还会联系上文生成的中文描述。

图 3.5 两种模型生成中文描述的比较。左侧为 LSTM 基础模型，右侧为基于 Attention 机制的模型。Attention 模型对场景的信息捕捉更灵敏，上图中捕捉到了街道上的车，而下图中捕捉到了柜子和冰箱。

图 3.6 两种模型生成中文描述的比较。左侧为 LSTM 基础模型，右侧为基于 Attention 机制的模型。Attention 模型描述错了数目。

3.5 两种模型的结果比较

图 3.5 是两种模型的对比，可以看出 Attention 的模型观察更细微全面，这是因为 Attention 的机制保证它可以“注意”到很多不同的元素。尽管基于 Attention 的 LSTM 模型具有上述优势，我们也看到它的缺点。比如，它不能始终保持很好的效果，有时候对数量的描述会出现偏差，如图 3.6；有时候也会将男人女人认错。所以 Attention 模型也不是很完美，也会犯一些常犯的错误，只是犯错的次数少一些。

另外，对于训练集中没出现过的场景，两种模型都表现都较差（见图 3.7），这种情况首要的任务是增加训练集数据量，提高训练数据的质量。

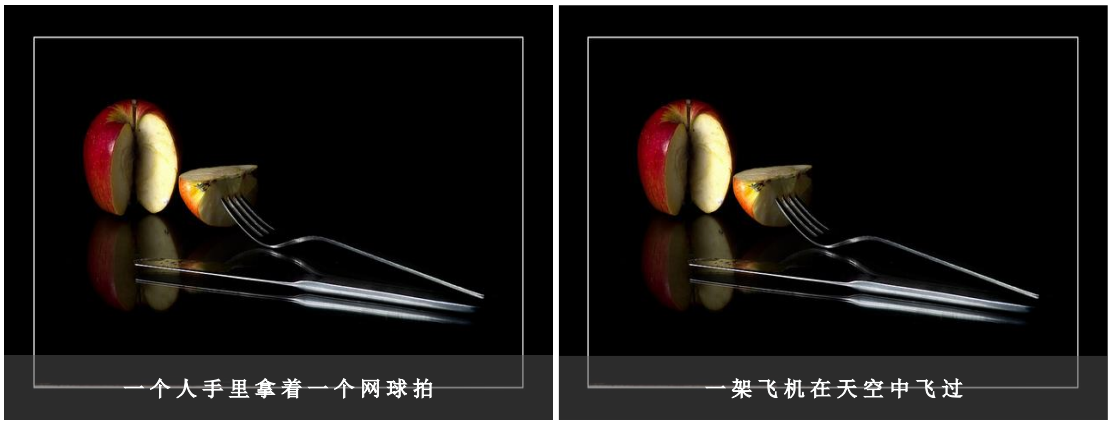

图 3.7 两种模型生成中文描述的比较。左侧为 LSTM 基础模型，右侧为基于 Attention 机制的模型。LSTM 基础模型将叉子看成了网球拍，而 Attention 则将其看成了飞机。

4 讨论

在本次项目中，我们成功运用 LSTM 实现了图片的中文描述生成，而且我们在大眼睛实验室的测试集指标排名[1]中排名靠前，截止 2017 年 6 月 8 日，我们在 CIDEr 排名中位列第三名(ID: 2016211096)。最好成绩是(0.676, 0.542, 0.431, 0.34, 0.513, 1.162)。

实验中，基于 Attention 机制的的 LSTM 模型训练速度很快，只花了 1~2 小时就达到了最佳结果，并且可以很好地捕捉图片较为完整的信息。不过，模型对某些陌生的图片却无法获得理想的描述，我们分析其原因可能是我们的训练集还

[1] 大眼睛实验室测试集指标排名：http://bigeye.au.tsinghua.edu.cn:12345/score?sort=5

不够大（9000 张图片，每张图片 3-5 句图片）。

基于我们的实验，我们对模型的改进如下建议：1）基于 Attention 机制的的 LSTM 模型的学习率在 0.0002 效果最佳；2）可以在 Attention 模型中引入 BeamSearch 的机制来生成测试集图片的中文描述；3）可以使用先进的优化算法，比如 RMSprop 算法代替 SGD 优化算法；可以使用衰减的学习率；4）可以尝试改进过拟合的问题，来提升模型生成的中文描述的质量。

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 P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.

3 K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine transla- tion. In EMNLP, 2014.

4 Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neu- ral machine translation by jointly learning to align and trans- late. arXiv:1409.0473 [cs.CL], September 2014.

5 Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual Attention[C]//International Conference on Machine Learning. 2015: 2048-2057.

6 Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

7 Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, and Ben- gio, Yoshua. How to construct deep recurrent neural networks. In ICLR, 2014.