第二次提交报告 Image Caption Generator

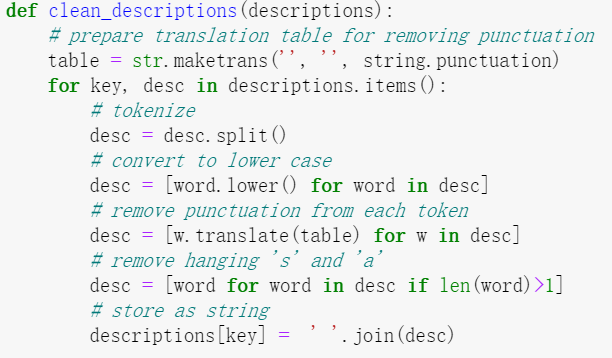
杨宝旭 罗迪 第2组

**一、模型主体流程：**

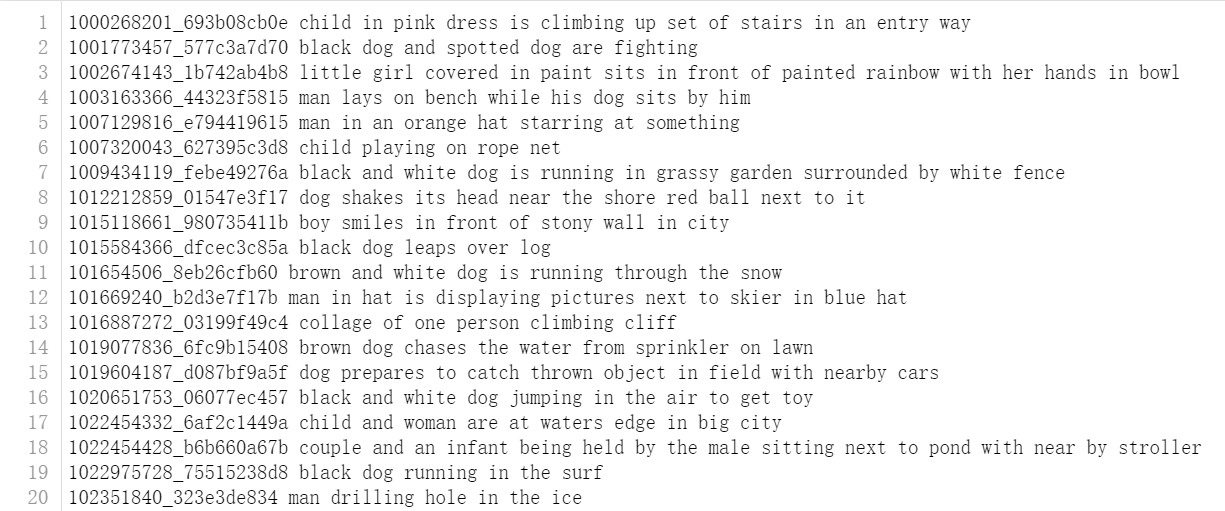
我们的图像字幕生成模型的代码框架总体可以分成四个部分：数据预处理、卷积提取图片特征、模型训练和模型评价。下面将从四个部分逐一分析我们的流程思路。

**1、数据预处理**

这部分主要是对文本部分的数据预处理，即每张图片的描述语句。Flicker8k数据集的图片与语句描述部分是分开的，且每张图片对应有五句不同的描述语句。为了方便模型训练，我们只选取其中的一句作为模型的训练语句，且对语句进行了适当的预处理：将所有字母小写、去除标点符号、去除单个的字母s和a：



然后将描述语句信息保存，格式是每一行分为图片名字和对应的描述语句，便于后续直接根据图像名称直接索引到对应的描述语句。



**2、卷积提取图片特征**

按照正常的流程，我们还需要训练一个卷积神经网络用来提取图片特征。但是训练一个效果较好的卷积神经网络需要规模庞大的数据集，还需要很多的时间来进行深度学习，这又是另一个领域的关注重点。在清楚地认识了我们的模型框架构成后，我们选择用现在已经有的卷积神经网络直接提取出我们的图片特征。

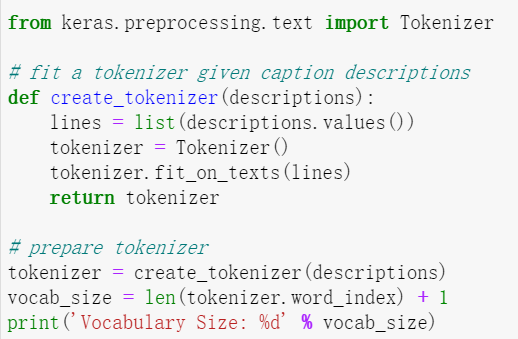
Keras的Applications包中提供了很多预训练好的卷积神经网络模型，使用时只需要调用并下载好模型的权重即可。我们最终选用的是在2014年ImageNet竞赛中获胜的牛津视觉几何组的VGG模型。



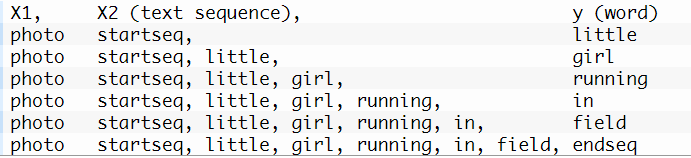
我们通过读取图片的名称加载每一张图片，VGG模型的predict函数返回图片的特征，它是一个形状为的三维数组，并把它以字典的形式保存在文件中。这样我们就可以通过图片的名称加读文件的方式，得到图片的特征数组，简单高效。

**3、模型训练**

首先是按照比例把数据集分割成训练集和测试集，然后依次读取出它们各自的描述语句和图片特征信息；随后是基于训练集和测试集的所有描述语句构建分词器Tokenizer，把每个单词对应一个数字，这样便于机器读入与输出：



另一个方面，我们的模型每次只输出一个单词，所以输入除了图像特征外，文本其实是一个序列，代表之前所有的描述语句。起始输入为startseq，模型输出一个单词后，再把startseq连同这个单词一起作为输入，以此类推，直到模型输出endseq结束。



最关键的是模型的架构部分。这里参考了Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge的论文，模型整体可以分为三个板块：图片特征提取部分、文本特征提取部分、综合特征提取并预测输出部分。

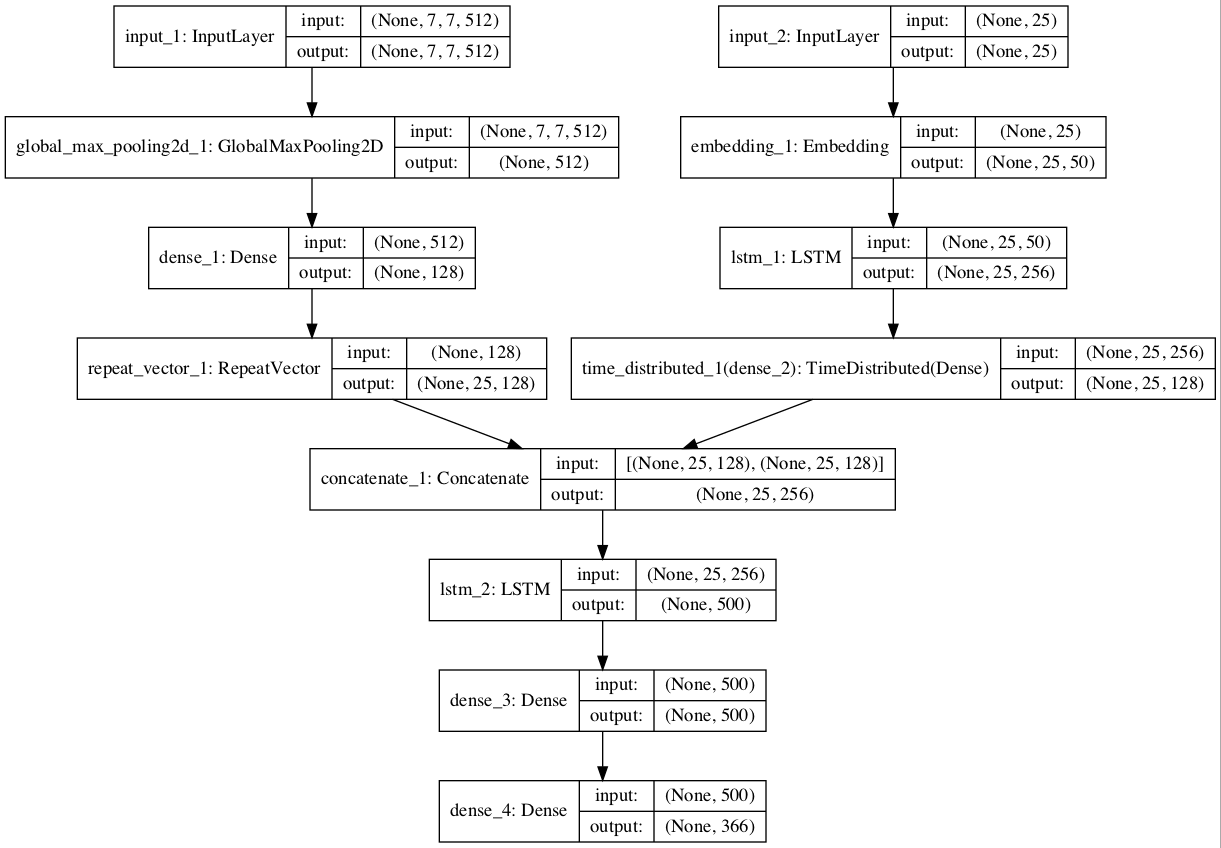
在图片特征提取部分，基于VGG16模型的输出作为输入，接下来是一个池化层，采用GlobalMaxPooling2D策略，随后是一个128的全连接层；

在文本特征提取部分，文本特征信息连接一个50维的词嵌入层，一个256维的LSTM层和128维的全连接层；

经过这两个板块的处理，图像信息和描述语句信息都被转换为了128维的向量，这时候将两者连接成一个256维的向量，再接一个500维的LSTM层和500维的全连接层，得到的输出表示整个词汇表上的概率分布。

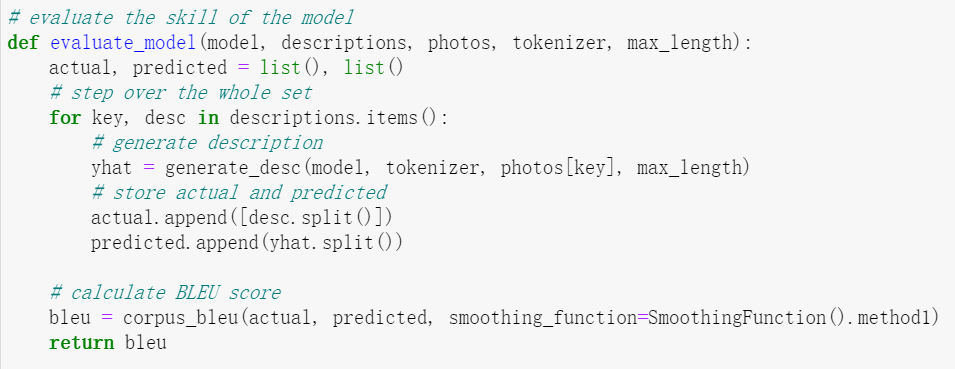
训练过程中使用softmax激活函数，在拟合网络时将分类交叉熵损失函数最小化。

最终的模型可视化效果如下：



**4、模型评价**

模型训练完成后，会对测试集的每张图片生成对应的描述语句，我们采用的是BLEU分数来评估模型的效果。在NLTK库的corpus\_bleu函数实现了对描述语句和参考语句之间的BLEU分数计算，越接近1代表效果越好。通过循环多次评估模型，我们可以得到方法模式下模型的平均得分，用来衡量此方法的模型训练效果。



**二、模型改进与优化：**

建立好模型的总体框架后，根据模型训练的结果和评估的分数，可以对深度学习网络的层数和结构进行调整，从而优化模型，得到更好的预测结果。

**（1）改变图像与文本特征的向量大小**

在最初的例子中，我们让图像特征和文本特征分别经过全连接层后输出为128维的向量再合并为256维的向量，用128维的向量对于信息而言可能过多/过少了，所以我们尝试了把输出向量从128维改成了64维和256维，最终是改成64维的效果更好。



**（2）改变文本特征的网络层数大小**

提取文本特征时，我们先采用一个50维的词嵌入层，后接一个256维的LSTM来实现，尝试了改变LSTM的大小来看对模型效果的影响，尝试了把LSTM由256个改为128个和把LSTM由一层改为两层，最终是添加一层256个单元的LSTM更好。



**（3）改变图片特征的池化层**

在模型定义的时候，我们首先采用的是最大池化的方法来提取图片特征，可以尝试改为平均池化，它可以更加综合地提取图片的特征。Keras通过GlobalMaxPooling2D和GlobalAveragePooling2D便捷地实现了两种不同的池化方式。除此之外，我们还尝试了不接池化层的做法（直接Flatten），来作为效果的对照。

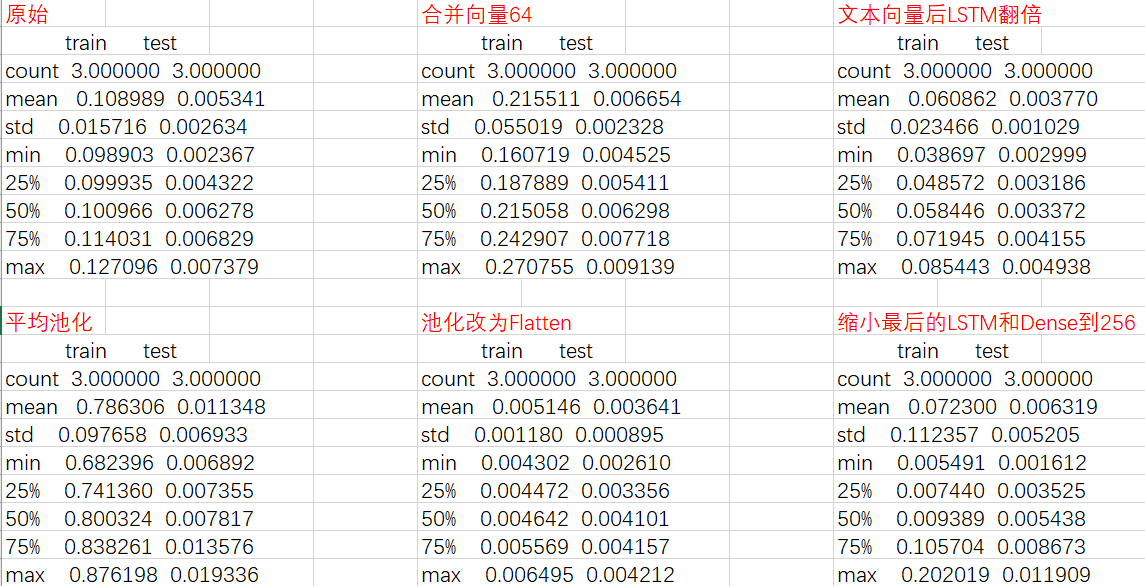


**（4）改变向量连接后产生输出的语言层大小**

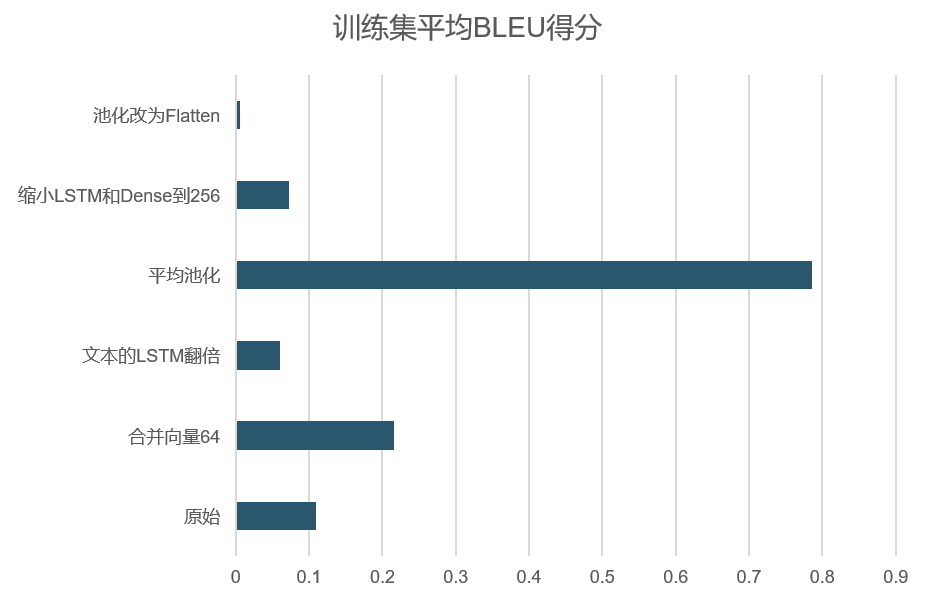
我们把两个向量连接后产生一个256维的图片文本特征向量，然后接一个500个单元的LSTM层和一个500个单元的全连接层来得到词汇表上的概率分布。在此基础上，我们尝试了把这个层的神经元个数由500个改为256个，来看看对模型效果的影响：

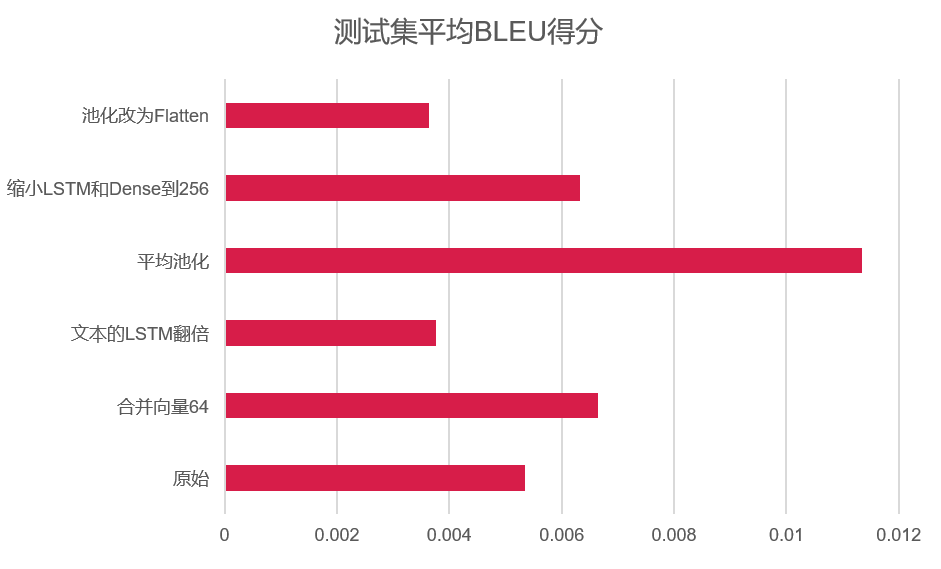


每次改动重复三次，计算BLEU分数的平均值，最终效果如下：



对训练集和测试集的BLEU分数分别做可视化，效果如下图：





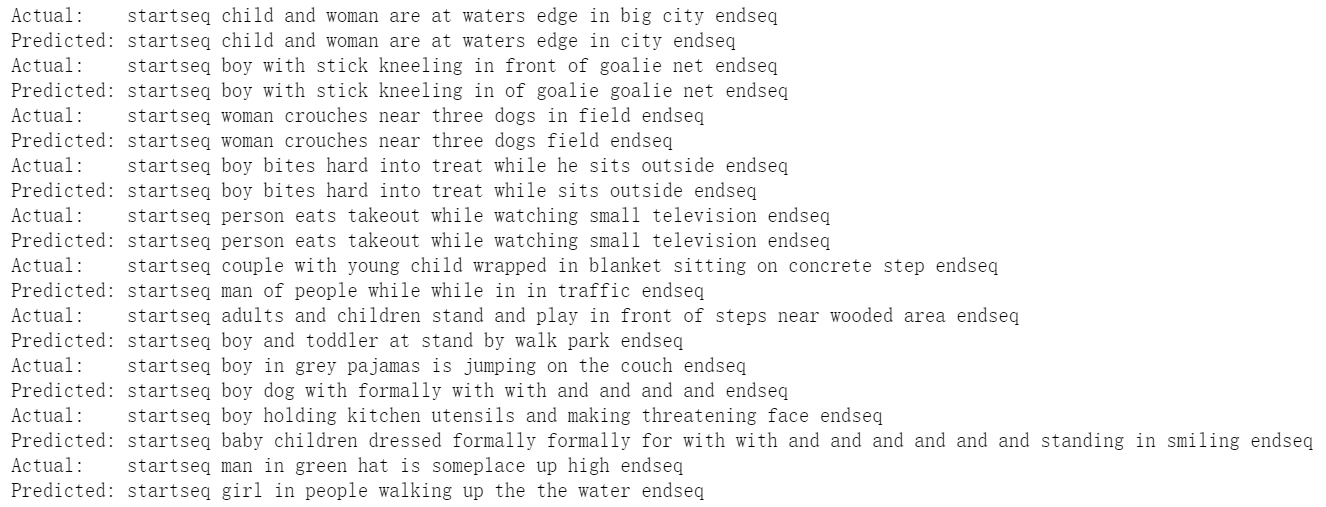
将池化层取消改为Flatten后，在训练集和测试集上效果均显著下降，说明在VGG模型提取图片特征的基础上再进行池化的重要性；

改变向量连接后语言层的LSTM和Dense大小到256后，在训练集上BLEU分数下降，但在测试集上效果提升，说明单元过多可能出现了过拟合现象；

平均池化的效果最好，在训练集上的BLEU分数接近0.8，在测试集上也有很好的效果，说明通过取平均值来池化比取最大值来池化效果更好；

合并向量由128改为64后，在训练集和测试集的效果都有一定的上升，说明不需要128维的向量就足以表达文本和图片的特征。

我们最终使用平均池化的方法训练了模型，并对训练集和测试集的各五条语句进行了预测输出，与真实的描述语句对比如下：



可见在训练集上，模型的预测输出与真实的描述语句基本一致，符合BLEU达到0.8的情况；即使在测试集上，模型也能够有效识别出一些显著特征，并试图输出一些语句，尽管效果还不太好，有待进一步提升。

**三、后期展望及下一步努力方向**

① 基于模型改进的策略组合

本阶段我们是在一个基准模型的基础上尝试了不同策略对于基准模型的效果改进，主要是有一个对照作用；后续可以考虑把本阶段尝试出来比较好的几个改进点组合在一起并进一步测试，以期达到更好的模型预测效果；

② 尝试更大的训练集与测试集

限于时间因素，本阶段没能够对整个数据集，以及一些更大更完备的数据集如MSCOCO进行测试；后续可以花费更多时间来提高数据量跑模型，因为更广泛的数据输入对于模型的可泛化性以及鲁棒性的提升会有很大的助力；

③ 模型框架引入更多的机制

目前参考的还是show and tell论文中的框架，在这篇论文后，很多作者提出了很多基于此框架的改进，如show, attend and tell论文中提到的attention机制等，后续可以考虑改进模型的框架，并实验这些机制对模型效果的改进程度；

④ 尝试在不同应用上的扩展

在实现了图像的字幕生成后，可以对现有模型的进行拓展，如尝试实现输入一个图片序列，返回一整段文字的看图说话；还有基于模型输出单词的知识图谱生成等。可能会在此模型基础上增加对图像的中的文本描述向量之间的逻辑关系的模型进行相关训练。