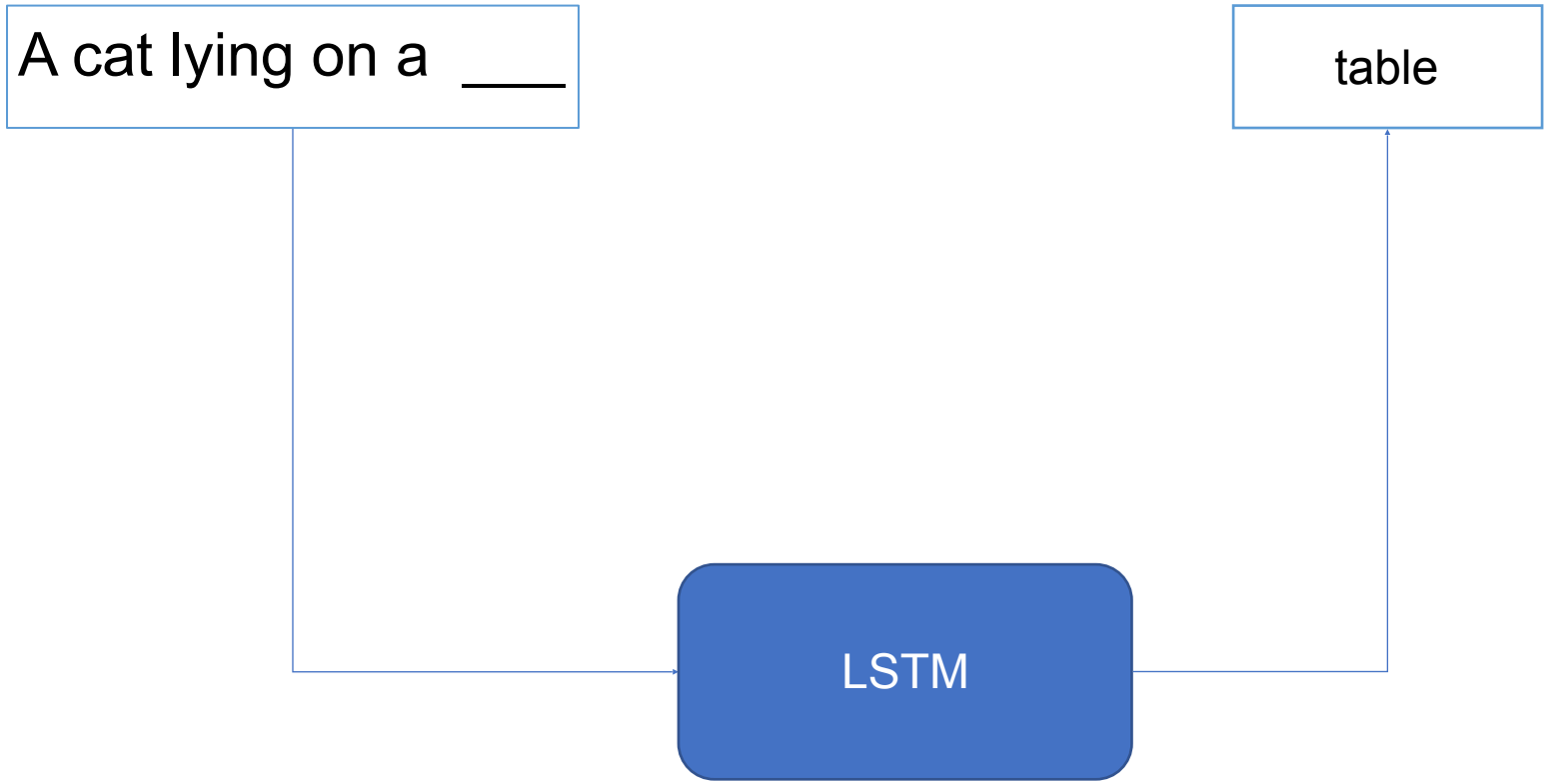
看图说话
(Image Captioning)

cat

CNN


A cat lying on a ____
table
LSTM

a cat lying on a table

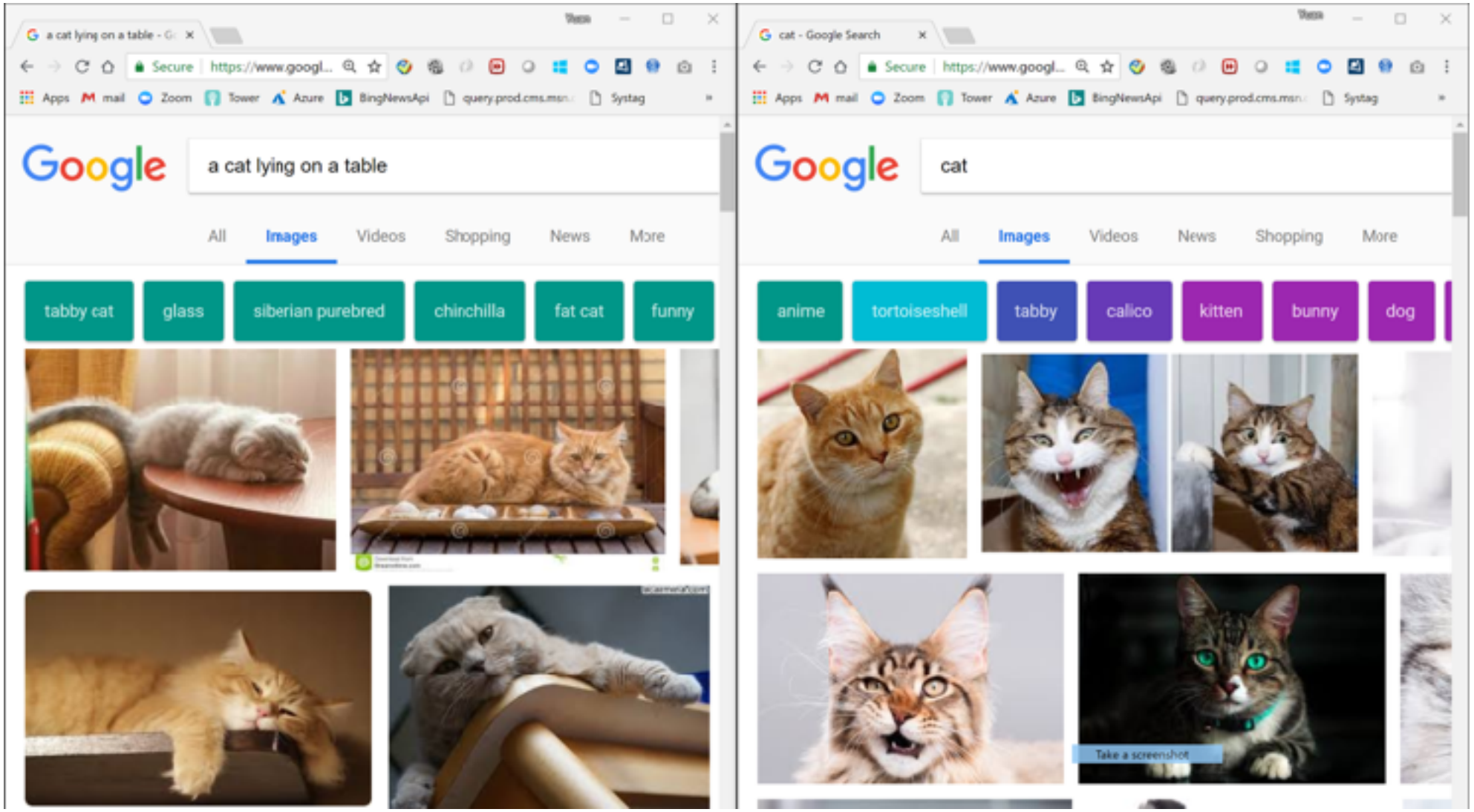
CNN + LSTM + DNN

看图说话

a cat lying on a table

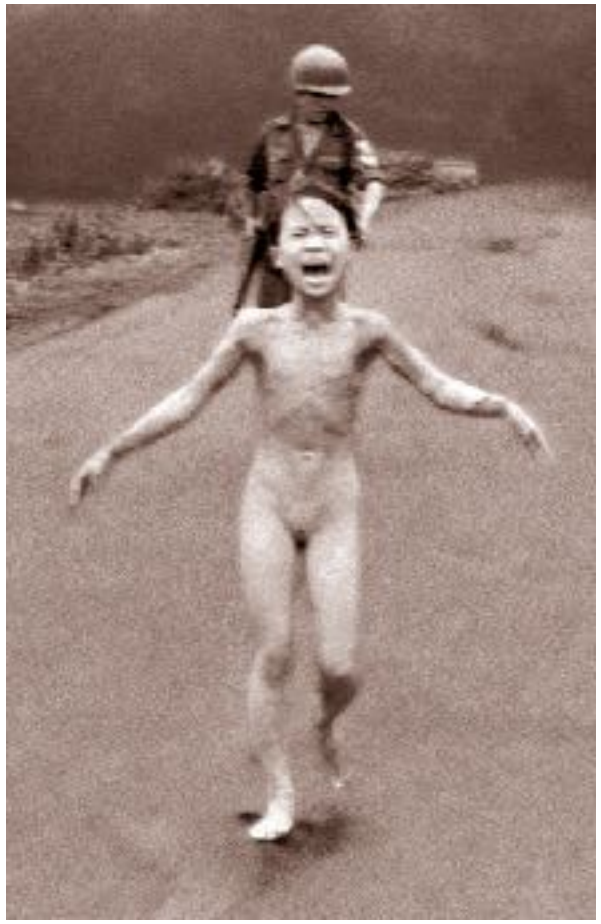
CNN + LSTM + DNN

应用领域：图像搜索


a cat lying on a table - G
Secure https://www.googl...
Apps mail Zoom Tower Azure BingNewsApi query.prod.cms.msn Systag
Google
a cat lying on a table
All Images Videos Shopping News More
tabby cat glass siberian purebred chinchilla fat cat funny
cat - Google Search
Secure https://www.googl...
Apps mail Zoom Tower Azure BingNewsApi query.prod.cms.msn Systag
Google
cat
All Images Videos News Shopping More
anime tortoiseshell tabby calico kitten bunny dog
Take a screenshot

应用领域：安全

应用领域: 鉴黄



涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)
- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)
- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)

- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)
- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)
- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

使用深度学习自动为图像生成标题

深度神经网络

a dog is running through the grass

涉猎的知识

- 数字图像处理
  - 图像读取
  - 图像缩放
  - 图像数据维度变换
- 自然语言处理
  - 文字清洗
  - 文字嵌入(Embedding)
- CNN(卷积神经网络)
  - 图像特征提取
  - 迁移学习(transfer learning)
- LSTM(递归神经网络)
  - 文字串(sequence)的特征提取
- DNN(深度神经网络)
  - 从图像特征和文字串(sequence)的特征预测下一个单词

实现方案

- Python 3.6.4:
  - Anaconda 安装包 64 位
  - windows: https://repo.continuum.io/archive/Anaconda3-5.1.0-Windows-x86_64.exe
  - mac: https://repo.continuum.io/archive/Anaconda3-5.1.0-MacOSX-x86_64.pkg
  - Linux:Anaconda3-5.1.0-Linux-x86_64.sh

- TensorFlow 1.8.0
  - conda install -c conda-forge tensorflow=1.8.0
- Keras 2.1.5
  - conda install -c conda-forge keras=2.1.5
- nltk 3.2.5
  - conda install -c anaconda nltk=3.2.5
- Pillow 5.0.0
  - conda install -c anaconda pillow=5.0.0
- OpenCV 3.4.1
  - conda install -c conda-forge opencv=3.4.1
- PyTorch 0.4.1
  - conda install pytorch=0.4.1 -c pytorch

数据集 (Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics, 2013.)

- Flickr8K
- 8000 图像, 每幅图5个标题, 描述图像里面的事物和事件
- 不包含著名人物和地点
- **分为3个集合: 6000个训练图像, 1000个开发图像, 1000个测试图像**

3637013_c675de77C5.j
pg

10815824_2997e03d76.
jpg

12830823_87d2654e31.
jpg

17273391_55cfc7d3d4.j
pg

19212715_20476497a3.
jpg

35506150_cbdb630f4f.j
pg

36422830_55c844bc2d.
jpg

41959070_838089137e.
jpg

42637986_135a9786a6.
jpg

42637987_865635ed16.
jpg

47871819_db55ac4699.
jpg

49553964_cee950f3ba.j
pg

50030244_02cd4de372.
jpg

53043785_c468d6f931.j
pg

54501196_a9ac9d65f2.j
pg

55473406_1d2271c1f2.j
pg

56489627_e1de43de34
.jpg

56494233_1824005879.
jpg

57417274_d55d34e93e
.jpg

57422853_b5f6366081.
jpg

数据示例

- A child in a pink dress is climbing up a set of stairs in an entry way .
- A girl going into a wooden building .
- A little girl climbing into a wooden playhouse .
- A little girl climbing the stairs to her playhouse .
- A little girl in a pink dress going into a wooden cabin

终极目标

自动生成图像英文标题, 与人类生成的标题们越相似越好

- A child in a pink dress is climbing up a set of stairs in an entry way .
- A girl going into a wooden building .
- A little girl climbing into a wooden playhouse .
- A little girl climbing the stairs to her playhouse .
- A little girl in a pink dress going into a wooden cabin

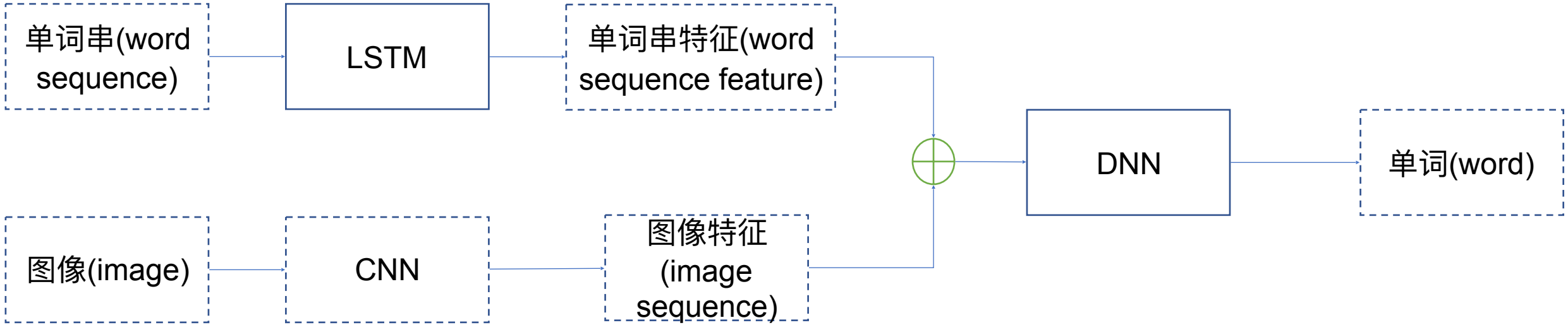
衡量两个句子的相似度(BLEU)
一个句子与其他几个句子的相似度(Corpus BLEU)

- BLEU: Bilingual Evaluation Understudy(双语评估替换)
- BLEU，全称为Bilingual Evaluation Understudy（双语评估替换），是一个比较候选文本翻译与其他一个或多个参考翻译的评价分数。
- 尽管BLEU一开始是为翻译工作而开发，但它也可以被用于评估自动生成文本的质量。

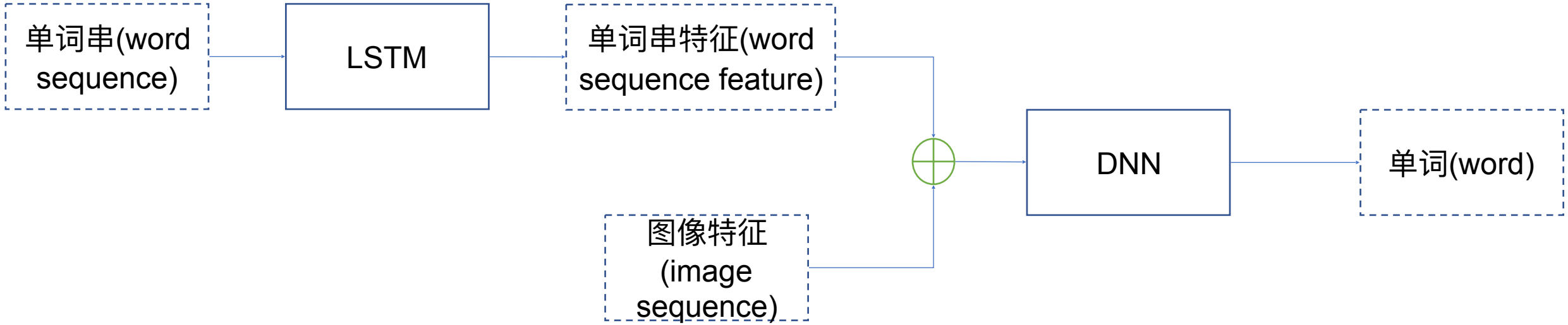
Python.BLEU.pdf

- 具体内容可以参考: https://cloud.tencent.com/developer/article/1042161 或者:

理想网络模型


单词串(word sequence)
LSTM
单词串特征(word sequence feature)
图像(image)
CNN
图像特征(image sequence)
DNN
单词(word)

简化网络模型


单词串(word sequence)
LSTM
单词串特征(word sequence feature)
图像特征 (image sequence)
DNN
单词(word)

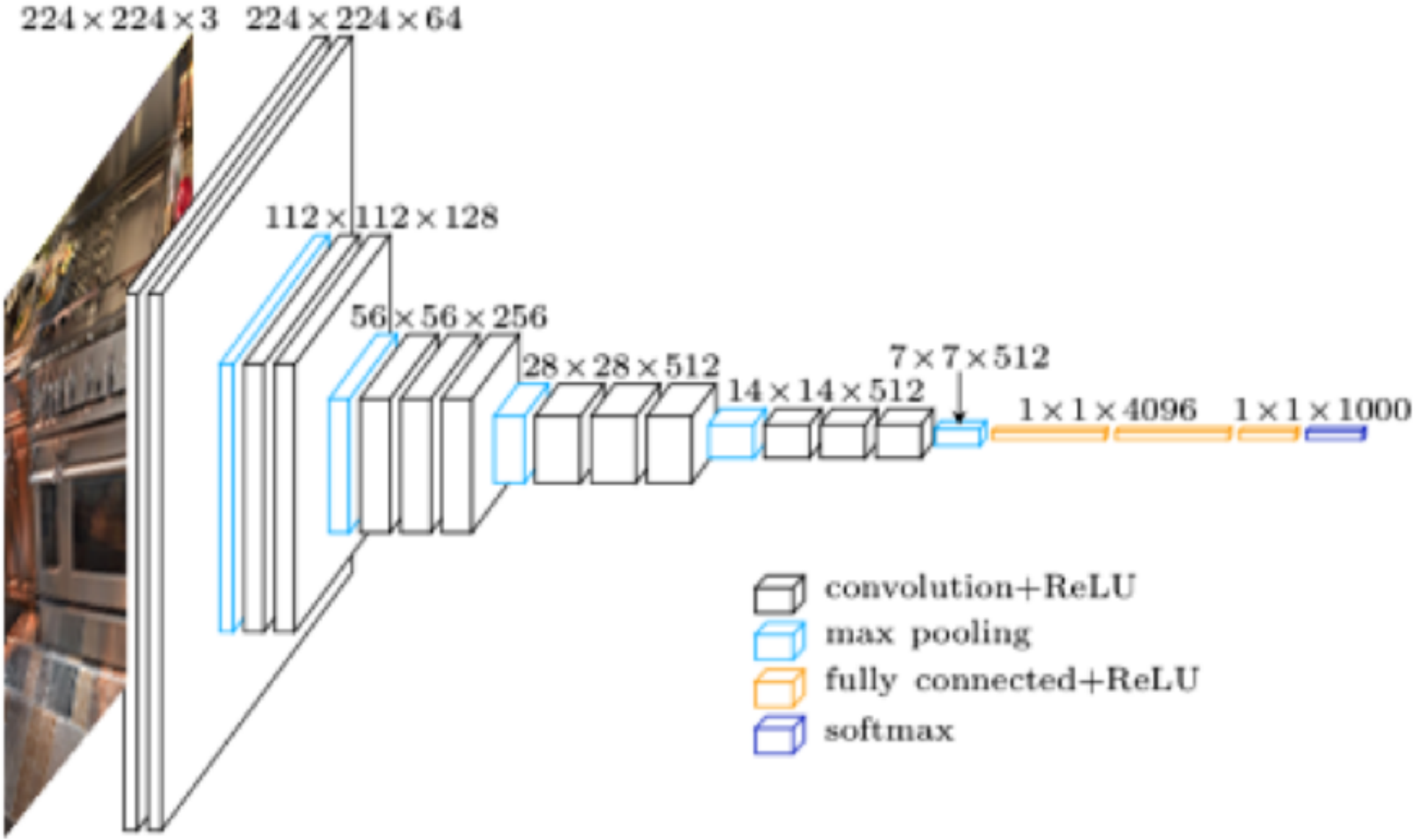
从图像(image)到图像特征(image feature)

[0.235, 0.457, 0.5685, ..., 0.956]

VGG16 (Very Deep Convolutional Networks for Large-Scale Visual Recognition)

- Pre-trained model: Oxford Visual Geometry Group赢得2014 ImageNet竞赛
- 用于图像分类, 将输入图像分为1000个类别

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

迁移学习(transfer learning)

- VGG16 CNN 原本的目标是分类, 基于ImageNet数据集进行训练, 训练所需的时间比较大,需要4个GPU训练3个星期左右
- 我们可以调整VGG16的网络结构为图像标题生成服务
- VGG16 的最后一层是将倒数第二层4096维的输出转为1000维的输出作为1000类别的分类概率
- 我们可以去除最后一层,将倒数第二层的4096维的输出作为图像标题生成模型的图像特征

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB imag)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

单词嵌入(Word Embedding)

- LSTM的输入是数值, 单词需要转换为数值才能使用LSTM, 最简单的方式是将单词转化为整数,每个单词都对应于一个整数. 但是这样的方式无法有效的表达单词直接的相关性, 例如:国王, 王后, 男,女
  如果整数10,11,200,300表示的话,无法体现出: 国王 – 男 + 女 ≈ 王后

- 单词嵌入是利用神经网络来学习单词的表达,使用一个向量而不是一个整数来表达一个单词. 向量提供了更大的信息量, 里面可以嵌入单词之间的关系, 更好的表达一个单词.

根据图像生成图像的标题

- 步骤:
  - 提取图像的特征(利用VGG16的修改模型)
  - 初始化图像标题为”startseq”
  - 循环如下步骤:
    - 将图像标题转换为整数数组,每一个标题的单词对应于唯一一个整数
    - 将图像特征和当前的图像标题作为输入, 预测标题的下一个单词, 假设单词为word1
    - 将word1添加到当前标题的结尾
    - 如果word1的值为”endseq”, 或者当前标题的长度达到了标题最大长度, 退出循环
  - 此刻的图像标题就是预测的值

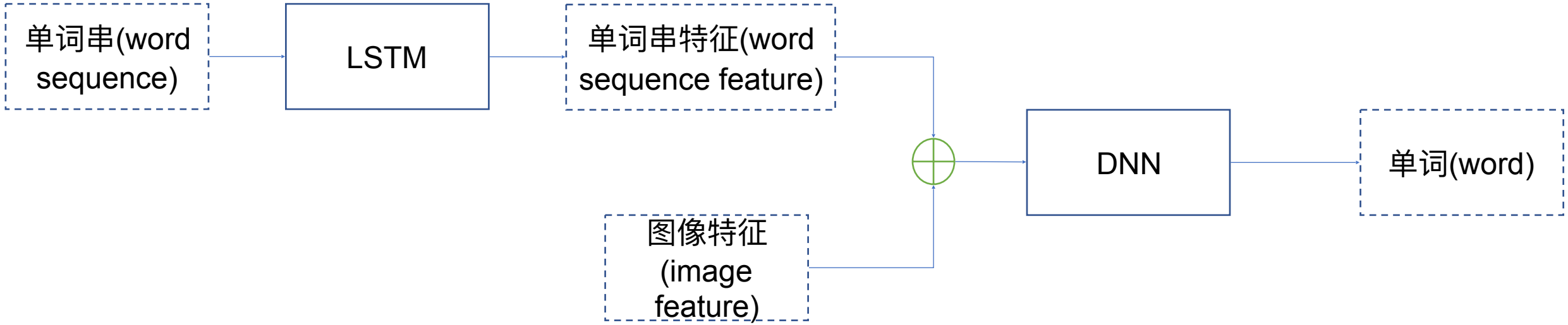
任务一: 使用keras创建VGG16定义的CNN网络结构

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

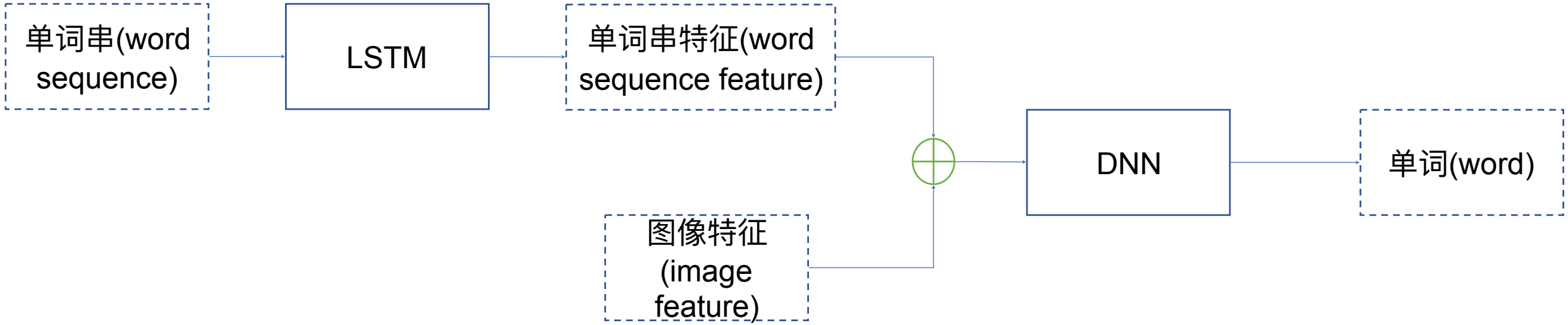
任务二: 将flicker8k的图像文件转为图像特征, 保存为字典pickle文件

- 从给定的VGG16网络结构文件和网络权值文件, 创建VGG16网络
- 修改网络结构(去除最后一层)
- 利用修改的网络结构,提取flicker8k数据集中所有图像的特征,使用字典存储, key为文件名(不带.jpg后缀), value为一个网络的输出
- 将字典保存为features.pkl文件(使用pickle库)

任务三: 完成create_tokenizer, create_input_data_for_one_image函数,用于产生如下网络结构的输入



任务三: 完成create_tokenizer, create_input_data_for_one_image函数,用于产生如下网络结构的输入



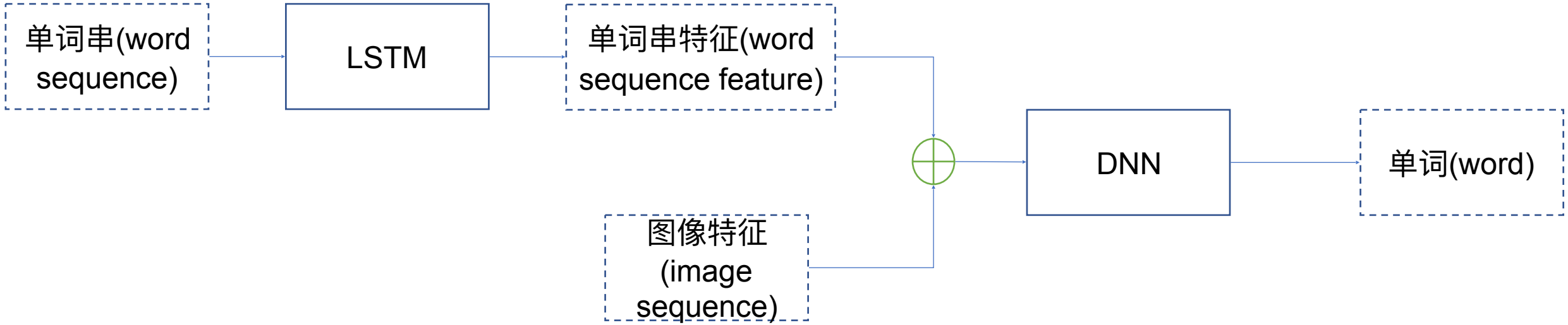
任务三: 完成create_input_data_for_one_image函数

为了训练LSTM, 训练数据中的每一个图像的每一个标题都需要被重新拆分为输入和输出部分. 如果标题为”a cat sits on the table”, 需要添加起始和结束标志, 变为 ‘startseq a cat sits on the table endseq’, 再从它产生如下训练数据序列:

图像输入	文字输入	输出
[0.234, 0.1124, …, 0.046]	startseq	a
[0.234, 0.1124, …, 0.046]	startseq a	cat
[0.234, 0.1124, …, 0.046]	startseq a cat	sits
[0.234, 0.1124, …, 0.046]	startseq a cat sits	on
[0.234, 0.1124, …, 0.046]	startseq a cat sits on	the
[0.234, 0.1124, …, 0.046]	startseq a cat sits on the	table
[0.234, 0.1124, …, 0.046]	startseq a cat sits on the table	endseq

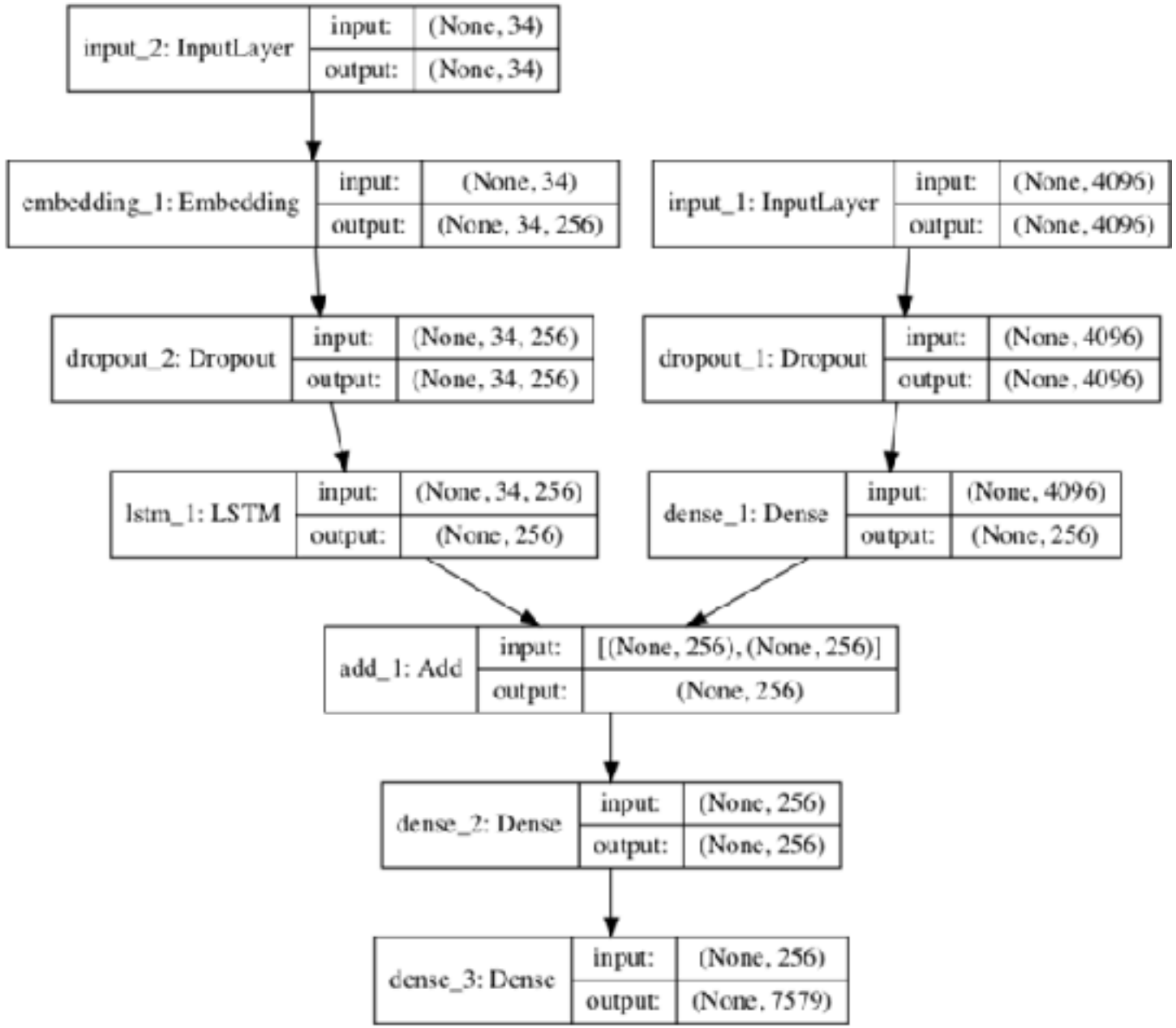
另外, 需要预处理单词,去掉 ‘s 和一些不需要的标点符号, 还需要将每一个单词转换为一个整数

任务四:构建自动产生图像标题的网络模型, 可以按照如下图的思路构建, 然后训练网络



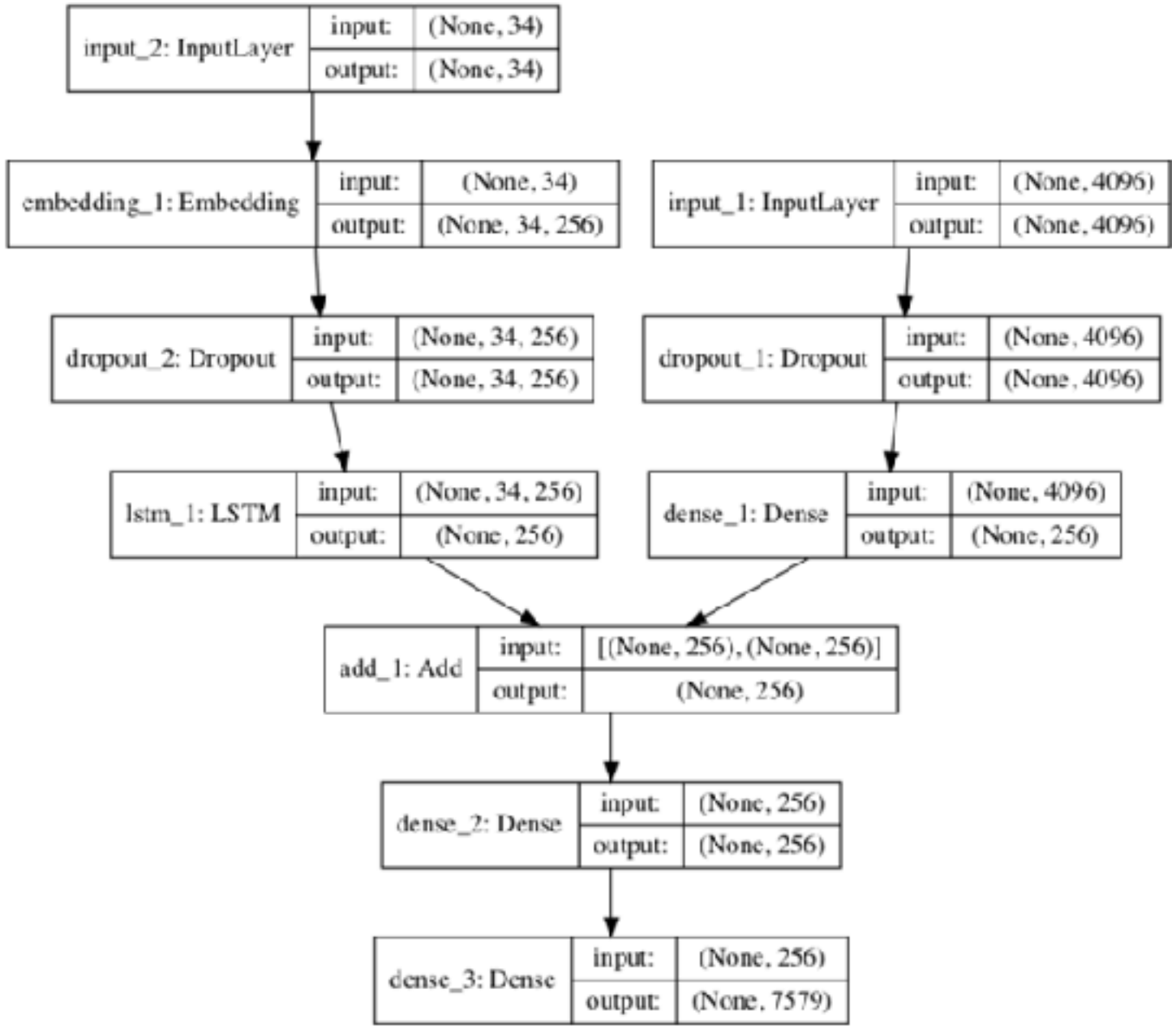
注意LSTM的第一层应该是一个嵌入层(embedding layer),用于将整数表达的单词转换为向量表达

使用交叉验证cross_validation来衡量不同结构的优劣


input_2: InputLayer
input: (None, 34)
output: (None, 34)
embedding_1: Embedding
input: (None, 34)
output: (None, 34, 256)
input_1: InputLayer
input: (None, 4096)
output: (None, 4096)
dropout_2: Dropout
input: (None, 34, 256)
output: (None, 34, 256)
dropout_1: Dropout
input: (None, 4096)
output: (None, 4096)
lstm_1: LSTM
input: (None, 34, 256)
output: (None, 256)
dense_1: Dense
input: (None, 4096)
output: (None, 256)
add_1: Add
input: [(None, 256), (None, 256)]
output: (None, 256)
dense_2: Dense
input: (None, 256)
output: (None, 256)
dense_3: Dense
input: (None, 256)
output: (None, 7579)

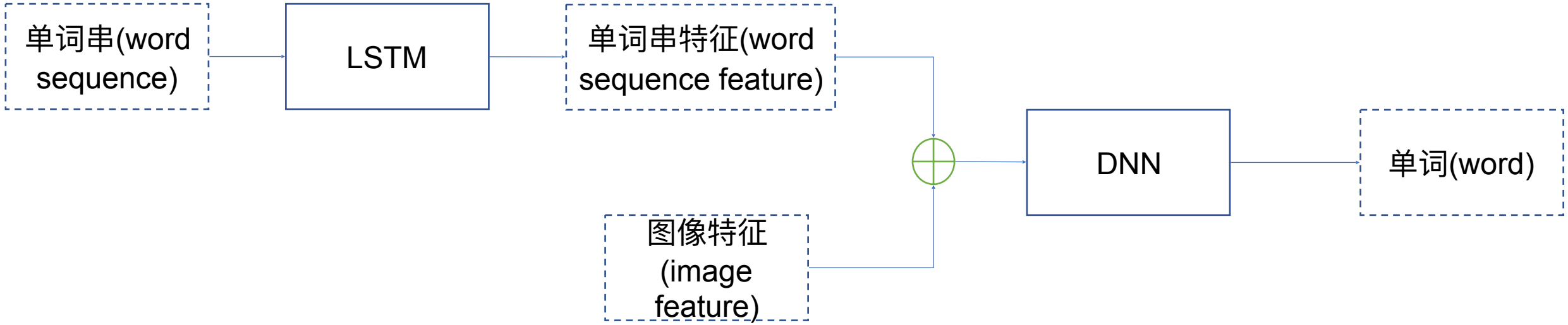
任务5

- 完成预测generate_caption代码
- 了解如何评价模型的性能


input_2: InputLayer
input: (None, 34)
output: (None, 34)
embedding_1: Embedding
input: (None, 34)
output: (None, 34, 256)
input_1: InputLayer
input: (None, 4096)
output: (None, 4096)
dropout_2: Dropout
input: (None, 34, 256)
output: (None, 34, 256)
dropout_1: Dropout
input: (None, 4096)
output: (None, 4096)
lstm_1: LSTM
input: (None, 34, 256)
output: (None, 256)
dense_1: Dense
input: (None, 4096)
output: (None, 256)
add_1: Add
input: [(None, 256), (None, 256)]
output: (None, 256)
dense_2: Dense
input: (None, 256)
output: (None, 256)
dense_3: Dense
input: (None, 256)
output: (None, 7579)

图像输入	文字输入	输出
[0.234, 0.1124, …, 0.046]	startseq	a
[0.234, 0.1124, …, 0.046]	startseq a	cat
[0.234, 0.1124, …, 0.046]	startseq a cat	sits
[0.234, 0.1124, …, 0.046]	startseq a cat sits	on
[0.234, 0.1124, …, 0.046]	startseq a cat sits on	the
[0.234, 0.1124, …, 0.046]	startseq a cat sits on the	table
[0.234, 0.1124, …, 0.046]	startseq a cat sits on the table	endseq

- 完成预测generate_caption代码



任务5:完成预测generate_caption代码，了解如何评价模型的性能

- 使用4个 corpus BLEU分数来评价模型在测试集上面的表现
- 你可以根据这个评价回头去修改你的网络结构，可以重新重新训练

The End

- 整个模型可以从头开始训练, 但是CNN的模型非常大, 如果每一次希望改变语言模型,都训练一遍CNN, 非常耗时
- 可以分开训练
- 可以将CNN部分预训练,作为图像特征提取器, 把每一副输入的图像转变为一组图像特征值
- 这样可以训练更快, 更加节约计算资源

使用预训练模式用于不同目的

- 从keras加载VGG预训练模型, 去掉最后一层 (原来最后一层的目的用于图像分类)
- 倒数第二层的输出可以认为是图像的特征数据 (1维, 4096长度的向量)
- VGG要求的输入图像为3通道,244像素宽, 244像素高