**TensorFlow｜3｜词嵌入**

1. 基本环境

|  |  |
| --- | --- |
| 程序名称 | myWordToVec.py |
| 硬件环境 | MacOSX |
| 软件环境 | Python3, Spyder, Anaconda |

1. 核心问题：利用词嵌入模型进行特征提取
2. 解决过程：

* 由于句子长短不一，所以需要确定最大句长度，才能构成统一的矩阵，所以第一步是画图确定最大句长
* 构建词嵌入
* 查看词向量
* 查看词表中的词
* 显示空间距离相近的词

1. 代码：

#找到合适的句长

import matplotlib.pyplot as plt

#matplotlib inline

plt.hist(numWords, 50)

plt.xlabel('Sequence Length')

plt.ylabel('Frequency')

plt.axis([0, 150, 0, 4000])

plt.show()

# 转换成词向量

import word2vec

word2vec.word2vec('/Users/xuyizhou/Desktop/corpusSegDone.txt', '/Users/xuyizhou/Desktop/corpusWord2Vec.bin', size=300,verbose=True)

model = word2vec.load('/Users/xuyizhou/Desktop/corpusWord2Vec.bin')

print (model.vectors)

#8000条评论 每条140个词 每个词一个索引 一个索引对应一行矩阵

import tensorflow as tf

import numpy as np

m=len(fileTrainSeg)

n=60

firstSen=np.zeros((m,n),dtype='int32')

#for j in range(len(fileTrainSeg)):

for j in range(m):

print(j)

#如果某评论词数小于60

if len(fileTrainSeg[j][0].split())<=60:

for i in range(len(fileTrainSeg[j][0].split())): index=np.where(model.vocab==fileTrainSeg[j][0].split()[i])[0]

if index>=0:

if index<=2545:

firstSen[j][i]=int(index)

else:

firstSen[j][i]=0

else:

#如果某评论词数大于60

for i in range(60): index=np.where(model.vocab==fileTrainSeg[j][0].split()[i])[0]

if index>=0:

if index<=2545:

firstSen[j][i]=int(index)

else:

firstSen[j][i]=0

print("索引矩阵：")

print(firstSen)

#得到每一行的词语数量

numWords=[]

for i in range(len(fileTrainSeg)):

counter = len(fileTrainSeg[i][0].split())

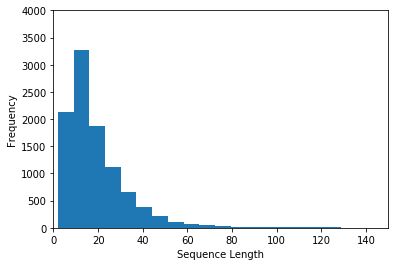
numWords.append(counter)

print('The total number of words in the files is', sum(numWords))

1. 结果：

**找到合适的句长**

得到句长的频率，可见句子长度大于60的频率很小，所以对于该数据集，最大句长设置为60。



**构建词向量**

word2vec.word2vec('/Users/xuyizhou/Desktop/corpusSegDone.txt', '/Users/xuyizhou/Desktop/corpusWord2Vec.bin', size=300,verbose=True)

Starting training using file /Users/xuyizhou/Desktop/corpusSegDone.txt

Vocab size: 4547

Words in train file: 342390

Alpha: 0.001557 Progress: 97.29% Words/thread/sec: 174.76k

**查看词向量**

[[ 0.08015625 0.08850128 -0.07670335 ..., -0.02626957 -0.03316621

0.0614953 ]

[ 0.03820458 -0.05974364 -0.05040395 ..., 0.03234643 -0.00140235

0.01892994]

[ 0.1301306 -0.00679784 0.01739487 ..., 0.0342424 -0.02699178

0.04948062]

...,

[ 0.02150656 -0.0800935 0.02247621 ..., 0.10062893 -0.01770805

-0.05905314]

[ 0.00276128 -0.04871397 -0.04063093 ..., 0.11567228 0.00949546

-0.06120371]

[ 0.05614121 -0.05768705 0.02015897 ..., 0.12129481 -0.02131688

-0.05464119]]

**查看词表中的词**

>>> print (model.vocab[1000])

更是

**显示空间距离相近的词**

>>> indexes = model.cosine(u'喜欢')

>>> for index in indexes[0]:

... print (model.vocab[index])

...

舒服

老公

帅气

推荐

合身

合适

气质

刚好

效果

好看