

# Python做文本情感分析之情感极性分析



作者 一个不知死活的胖子 (/u/b04c0d6732ff) (+关注)

2016.08.30 05:36\* 字数 2745 阅读 9287 评论 29 喜欢 35

(/u/b04c0d6732ff)

## 导语

「NLP」最为目前及其火热的一个领域，已经逐渐渗透进越来越多产业的各项业务中，不知死活的胖子决定对常用的应用功能挨个进行尝试，死活不论……

## 0. 介绍

「情感极性分析」是对带有感情色彩的主观性文本进行分析、处理、归纳和推理的过程。按照处理文本的类别不同，可分为基于新闻评论的情感分析和基于产品评论的情感分析。其中，前者多用于舆情监控和信息预测，后者可帮助用户了解某一产品在大众心目中的口碑。

目前常见的情感极性分析方法主要是两种：基于情感词典的方法和基于机器学习的方法。

## 1. 基于情感词典的文本情感极性分析

笔者是通过情感打分的方式进行文本情感极性判断， $score > 0$  判断为正向， $score < 0$  判断为负向。

### 1.1 数据准备

#### 1.1.1 情感词典及对应分数

词典来源于BosonNLP数据下载 (<http://bosonnlp.com/dev/resource>)的情感词典，来源于社交媒体文本，所以词典适用于处理社交媒体的情感分析。

词典把所有常用词都打上了唯一分数有许多不足之处。

- 之一，不带情感色彩的停用词会影响文本情感打分。在
- 之二，由于中文的博大精深，词性的多变成为了影响模型准确度的重要原因。一种情况是同一个词在不同的语境下可以是代表完全相反的情感意义，用笔者模型预测偏差最大的句子为例（来源于朋友圈文本）：

有车一族都用了这个宝贝，后果很严重 哦[偷笑][偷笑][偷笑]1，交警工资估计会打5折，没有超速罚款了[呲牙][呲牙][呲牙]2，移动联通公司大幅度裁员，电话费少了[呲牙][呲牙][呲牙]3，中石化中石油裁员2成，路痴不再迷路，省油[悠闲][悠闲][悠闲]5，保险公司裁员2成，保费折上折2成，全国通用[憨笑][憨笑][憨笑]买不买你自己看着办吧[调皮][调皮][调皮]

里面严重等词都是表达的相反意思，甚至整句话一起表示相反意思，不知死活的笔者还没能深入研究如何用词典的方法解决这类问题，但也许可以用机器学习的方法让神经网络进行学习能够初步解决这一问题。

另外，同一个词可作多种词性，那么情感分数也不应相同，例如：

这部电影真垃圾

垃圾 分类



很明显在第一句中 垃圾 表现强烈的贬义，而在第二句中表示中性，单一评分对于这类问题的分类难免有失偏颇。

1.1.2 否定词词典

否定词的出现将直接将句子情感转向相反的方向，而且通常效用是叠加的。常见的否定词：不、没、无、非、莫、弗、勿、毋、未、否、别、無、休、难道 等。

1.1.3 程度副词词典

既是通过打分的方式判断文本的情感正负，那么分数绝对值的大小则通常表示情感强弱。既涉及到程度强弱的问题，那么程度副词的引入就是势在必行的。词典可从《知网》情感分析用词语集（beta版）(<http://www.keenage.com/download/sentiment.rar>)下载。词典内数据格式可参考如下格式，即共两列，第一列为程度副词，第二列是程度数值，> 1表示强化情感，< 1表示弱化情感。

```
百分之百,2
倍加,2
备至,2
不得了,2
不堪,2
不可开交,2
不亦乐乎,2
不折不扣,2
彻头彻尾,2
充分,2
到头,2
地地道道,2
非常,2
极,2
极度,2
极端,2
极其,2
极为,2
截然,2
尽,2
惊人地,2
绝,2
绝顶,2
绝对,2
绝对化,2
刻骨,2
酷,2
满,2
满贯,2
满心,2
莫大,2
奇,2
入骨,2
甚为,2
```

程度副词词典

1.1.4 停用词词典

中科院计算所中文自然语言处理开放平台发布了有1208个停用词的中文停用词表(<http://www.datatang.com/data/43894>)，也有其他不需要积分的下载途径(<http://www.hicode.cn/download/view-software-13784.html>)。

1.2 数据预处理

1.2.1 分词

即将句子拆分为词语集合，结果如下：  
e.g. 这样的/酒店/配/这样的/价格/还算/不错

Python常用的分词工具：

- 结巴分词 Jieba
- Pymmsseg-cpp



- Loso
- smallseg

```
from collections import defaultdict
import os
import re
import jieba
import codecs

"""
1. 文本切割
"""

def sent2word(sentence):
    """
    Segment a sentence to words
    Delete stopwords
    """
    segList = jieba.cut(sentence)
    segResult = []
    for w in segList:
        segResult.append(w)

    stopwords = readLines('stop_words.txt')
    newSent = []
    for word in segResult:
        if word in stopwords:
            # print "stopword: %s" % word
            continue
        else:
            newSent.append(word)

    return newSent
```

在此笔者使用**Jieba**进行分词。

### 1.2.2 去除停用词

遍历所有语料中的所有词语，**删除其中的停用词**

e.g. 这样/的/酒店/配/这样/的/价格/还算/不错

→ 酒店/配/价格/还算/不错

## 1.3 构建模型

### 1.3.1 将词语分类并记录其位置

将句子中各类词分别存储并标注位置。



```

"""
2. 情感定位
"""
def classifyWords(wordDict):
    # (1) 情感词
    senList = readLines('BosonNLP_sentiment_score.txt')
    senDict = defaultdict()
    for s in senList:
        senDict[s.split(' ')[0]] = s.split(' ')[1]
    # (2) 否定词
    notList = readLines('notDict.txt')
    # (3) 程度副词
    degreeList = readLines('degreeDict.txt')
    degreeDict = defaultdict()
    for d in degreeList:
        degreeDict[d.split(',')[0]] = d.split(',')[1]

    senWord = defaultdict()
    notWord = defaultdict()
    degreeWord = defaultdict()

    for word in wordDict.keys():
        if word in senDict.keys() and word not in notList and word not in degreeDict.keys():
            senWord[wordDict[word]] = senDict[word]
        elif word in notList and word not in degreeDict.keys():
            notWord[wordDict[word]] = -1
        elif word in degreeDict.keys():
            degreeWord[wordDict[word]] = degreeDict[word]
    return senWord, notWord, degreeWord

```

### 1.3.2 计算句子得分

在此，简化的情感分数计算逻辑：**所有情感词语组的分数之和**

定义一个**情感词语组**：两情感词之间的所有否定词和程度副词与这两情感词中的后一情感词构成一个情感词组，即 `notWords + degreeWords + sentiWords`，例如 不是很交好，其中 不是 为否定词，很 为程度副词，交好 为情感词，那么这个情感词语组的分数为：

$$\text{finalSentiScore} = (-1)^1 * 1.25 * 0.747127733968$$

其中 1 指的是一个否定词，1.25 是程度副词的数值，0.747127733968 为 交好 的情感分数。

伪代码如下：

```

finalSentiScore = (-1) ^ (num of notWords) * degreeNum * sentiScore
finalScore = sum(finalSentiScore)

```



```
"""
3. 情感聚合
"""
def scoreSent(senWord, notWord, degreeWord, segResult):
    W = 1
    score = 0
    # 存所有情感词的位置的列表
    senLoc = senWord.keys()
    notLoc = notWord.keys()
    degreeLoc = degreeWord.keys()
    senloc = -1
    # notloc = -1
    # degreeLoc = -1

    # 遍历句中所有单词segResult, i为单词绝对位置
    for i in range(0, len(segResult)):
        # 如果该词为情感词
        if i in senLoc:
            # loc为情感词位置列表的序号
            senloc += 1
            # 直接添加该情感词分数
            score += W * float(senWord[i])
            # print "score = %f" % score
            if senloc < len(senLoc) - 1:
                # 判断该情感词与下一情感词之间是否有否定词或程度副词
                # j为绝对位置
                for j in range(senLoc[senloc], senLoc[senloc + 1]):
                    # 如果有否定词
                    if j in notLoc:
                        W *= -1
                    # 如果有程度副词
                    elif j in degreeLoc:
                        W *= float(degreeWord[j])
            # i定位至下一个情感词
            if senloc < len(senLoc) - 1:
                i = senLoc[senloc + 1]
    return score
```

1.4 模型评价

将600多条朋友圈文本的得分排序后做出散点图：



Score Distribution

其中大多数文本被判为正向文本符合实际情况，且绝大多数文本的情感得分的绝对值在10以内，这是因为笔者在计算一个文本的情感得分时，以句号作为一句话结束的标志，在一句话内，情感词语组的分数累加，如若一个文本中含有多句话时，则取其**所有句子情感得分的平均值**。

然而，这个模型的**缺点与局限性**也非常明显：



- 首先，段落的得分是其所有句子得分的平均值，这一方法并不符合实际情况。正如文章中先后段落有重要性大小之分，一个段落中前后句子也同样有重要性的差异。
- 其次，有一类文本使用贬义词来表示正向意义，这类情况常出现与**宣传**文本中，还是那个例子：  
有车一族都用了这个宝贝，后果很严重哦[偷笑][偷笑][偷笑]1，交警工资估计会打5折，没有超速罚款了[呲牙][呲牙][呲牙]2，移动联通公司大幅度裁员，电话费少了[呲牙][呲牙][呲牙]3，中石化中石油裁员2成，路痴不再迷路，省油[悠闲][悠闲][悠闲]5，保险公司裁员2成，保费折上折2成，全国通用[憨笑][憨笑][憨笑]买不买你自己看着办吧[调皮][调皮][调皮]2980元轩轾魔镜带回家，推广还有返利[得意]  
Score Distribution中得分小于 -10 的几个文本都是与这类情况相似，这也许需要**深度学习**的方法才能有效解决这类问题，普通机器学习方法也是很难的。
- 对于正负向文本的判断，该算法忽略了很多其他的否定词、程度副词和情感词搭配的情况；用于判断情感强弱也过于简单。

总之，这一模型只能用做BENCHMARK...

## 2. 基于机器学习的文本情感极性分析

### 2.1 还是数据准备

#### 2.1.1 停用词

(同1.1.4)

#### 2.1.2 正负向语料库

来源于有关中文情感挖掘的酒店评论语料 (<http://www.datatang.com/data/11936>)，其中正向7000条，负向3000条（笔者是不是可以认为这个世界还是充满着满满的善意呢...），当然也可以参考情感分析资源（转）(<http://www.cnblogs.com/finesite/p/3381803.html>)使用其他语料作为训练集。

#### 2.1.3 验证集

Amazon上对iPhone 6s的评论，来源已不可考.....

### 2.2 数据预处理

#### 2.2.1 还是要分词

(同1.2.1)



```
import numpy as np
import sys
import re
import codecs
import os
import jieba
from gensim.models import word2vec
from sklearn.cross_validation import train_test_split
from sklearn.externals import joblib
from sklearn.preprocessing import scale
from sklearn.svm import SVC
from sklearn.decomposition import PCA
from scipy import stats
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.optimizers import SGD
from sklearn.metrics import f1_score
from bayes_opt import BayesianOptimization as BO
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

def parseSent(sentence):
    seg_list = jieba.cut(sentence)
    output = ''.join(list(seg_list)) # use space to join them
    return output
```

### 2.2.2 也要去除停用词

( 同1.2.2 )

### 2.2.3 训练词向量

**( 重点来了！ ) 模型的输入需是数据元组，那么就需要将每条数据的词语组合转化为一个数值向量**

常见的转化算法有但不仅限于如下几种：

( 请原谅不知死活的胖子直接用展示的ppt截图作说明，没错，我就是懒，你打我呀 )

- **Bag of Words**



## Bag of Words (BOW)

### One-hot Representation

向量中每个分量表示词典中对应单词在文档中出现的次数

1. Bob likes to play basketball, Jim likes too.
2. Bob also likes to play football games.

构造词典

**Dict** = {1: Bob, 2: like, 3: to, 4: play, 5: basketball, 6: also, 7: football, 8: games, 9: Jim, 10: too}

**Vec1** = [1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

**Vec2** = [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

### 缺陷

“词汇鸿沟”现象：稀疏方式存储，其独立性假设不太符合语言文字实际分布情况，忽略了单词间的语法和顺序，无法了解单词间的关联程度

### 解决

- 采用SCPCD抽取整个短语
- 采用高阶（2阶以上）统计语言模型，例如bigram、trigram等

Bag of Words

#### • TF-IDF

## TF-IDF

如果某个词在一篇文章中出现的频率高，且在其他文章中出现的频率低，则认为该词具有很好的类别区分能力，赋予更高权重

**TF (Term Frequency)**：给定词语在该文档中出现的次数

**IDF (Inverse Document Frequency)**：词语普遍重要性的度量

### 缺陷

单纯以词频作为单词重要性的度量，对于位置没有敏感性

### 解决

人为添加权重

例如：第一段和最后一段赋予更高权重；每段第一句赋予更高权重

TF-IDF

#### • Word2Vec





# Word2Vec

词向量空间上的相似度可以用来表示文本语义上的相似度，常用于词语聚类、找同义词、词性分析等

INPUT PRODUCTION OUTPUT



**CBOW**  
(Continuous Bag-of-Words Model)

INPUT PRODUCTION OUTPUT



**Skip-Gram**  
(with )

**CBOW**：根据上下文单词预测位置t的单词

$$P(w_t | w_{t-k}, w_{t-(k-1)} \dots, w_{t-1}, w_{t+1}, w_{t+2} \dots, w_{t+k})$$

**Skip-Gram**：用于预测位置t的单词的上下文单词

$$P(w_i | w_t), \text{ 其中 } t - c \leq i \leq t + c \text{ 且 } i \neq t$$

**特点**

- 可进行向量的加法组合运算 (Additive Compositionality)  
例如： $\text{vec('king')} - \text{vec('man')} + \text{vec('woman')} = \text{vec('queen')}$
- 训练高效

Word2Vec

在此笔者选用Word2Vec将语料转化成向量，具体步骤可参考笔者的文章问答机器人的Python分类器 (<http://www.jianshu.com/p/8acb99b6671b>)。



```

def getWordVecs(wordList):
    vecs = []
    for word in wordList:
        word = word.replace('\n', '')
        try:
            vecs.append(model[word])
        except KeyError:
            continue
    # vecs = np.concatenate(vecs)
    return np.array(vecs, dtype = 'float')

def buildVecs(filename):
    posInput = []
    with open(filename, "rb") as txtfile:
        # print txtfile
        for lines in txtfile:
            lines = lines.split('\n ')
            for line in lines:
                line = jieba.cut(line)
                resultList = getWordVecs(line)
                # for each sentence, the mean vector of all its vectors is used to represent
                if len(resultList) != 0:
                    resultArray = sum(np.array(resultList))/len(resultList)
                    posInput.append(resultArray)

    return posInput

# load word2vec model
model = word2vec.Word2Vec.load_word2vec_format("corpus.model.bin", binary = True)
# txtfile = [u'标准间太差房间还不如3星的而且设施非常陈旧.建议酒店把老的标准间从新改善.', u'在这个西
posInput = buildVecs('pos.txt')
negInput = buildVecs('neg.txt')

# use 1 for positive sentiment, 0 for negative
y = np.concatenate((np.ones(len(posInput)), np.zeros(len(negInput))))

X = posInput[:]
for neg in negInput:
    X.append(neg)
X = np.array(X)

```

### 2.2.4 标准化

虽然笔者觉得在这一问题中，标准化对模型的准确率影响不大，当然也可以尝试其他的标准化的方法。

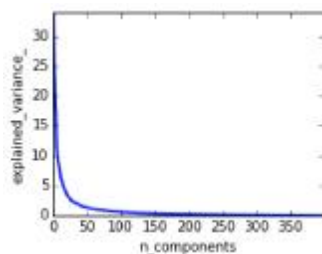
```

# standardization
X = scale(X)

```

### 2.2.5 降维

根据PCA结果，发现前100维能够cover 95%以上的variance。



PCA



```
# PCA
# Plot the PCA spectrum
pca.fit(X)
plt.figure(1, figsize=(4, 3))
plt.clf()
plt.axes([.2, .2, .7, .7])
plt.plot(pca.explained_variance_, linewidth=2)
plt.axis('tight')
plt.xlabel('n_components')
plt.ylabel('explained_variance_')

X_reduced = PCA(n_components = 100).fit_transform(X)
```

## 2.3 构建模型

### 2.3.1 SVM (RBF) + PCA

SVM (RBF)分类表现更为宽松，且使用PCA降维后的模型表现有明显提升，misclassified多为负向文本被分类为正向文本，其中  $AUC = 0.92$ ， $KSValue = 0.7$ 。关于SVM的调参可以参考笔者的另一篇文章Python利用Gaussian Process对Hyperparameter进行调参 (<http://www.jianshu.com/p/90e6abdeb4f2>)

```
"""
2.1 SVM (RBF)
    using training data with 100 dimensions
"""

clf = SVC(C = 2, probability = True)
clf.fit(X_reduced_train, y_reduced_train)

print 'Test Accuracy: %.2f' % clf.score(X_reduced_test, y_reduced_test)

pred_probab = clf.predict_proba(X_reduced_test)[:,-1]
print "KS value: %f" % KSmetric(y_reduced_test, pred_probab)[0]

# plot ROC curve
# AUC = 0.92
# KS = 0.7
fpr,tpr,_ = roc_curve(y_reduced_test, pred_probab)
roc_auc = auc(fpr,tpr)
plt.plot(fpr, tpr, label = 'area = %.2f' % roc_auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.legend(loc = 'lower right')
plt.show()

joblib.dump(clf, "SVC.pkl")
```

### 2.3.2 MLP

MLP相比于SVM (RBF)，分类更为严格，PCA降维后对模型准确率影响不大，misclassified多为正向文本被分类为负向，其实是更容易overfitting，原因是语料过少，其实用神经网络未免有些小题大做， $AUC = 0.91$ 。



```
"""
2.2 MLP
    using original training data with 400 dimensions
"""
model = Sequential()
model.add(Dense(512, input_dim = 400, init = 'uniform', activation = 'tanh'))
model.add(Dropout(0.5))
model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(32, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation = 'sigmoid'))

model.compile(loss = 'binary_crossentropy',
              optimizer = 'adam',
              metrics = ['accuracy'])

model.fit(X_train, y_train, nb_epoch = 20, batch_size = 16)
score = model.evaluate(X_test, y_test, batch_size = 16)
print ('Test accuracy: ', score[1])

pred_probab = model.predict(X_test)
# print "KS value: %f" % KSmetric(y_reduced_test, pred_probab)[0]

# plot ROC curve
# AUC = 0.91
fpr,tpr,_ = roc_curve(y_test, pred_probab)
roc_auc = auc(fpr,tpr)
plt.plot(fpr, tpr, label = 'area = %.2f' % roc_auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.legend(loc = 'lower right')
plt.show()
```

## 2.4 模型评价

- 实际上，第一种方法中的第二点缺点依然存在，但相比于基于词典的情感分析方法，基于机器学习的方法**更为客观**
- 另外由于训练集和测试集分别来自**不同领域**，所以有理由认为训练集不够充分，未来可以考虑扩充训练集以提升准确率。

📖 我爱机器学习 (/nb/5226241)

举报文章 © 著作权归作者所有



一个不知死活的胖子 (/u/b04c0d6732ff)

写了 6287 字，被 69 人关注，获得了 47 个喜欢  
(/u/b04c0d6732ff)

+ 关注

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持

♡ 喜欢 (/sign\_in) | 35



更多分享

(http://cwb.assets.jianshu.io/notes/images/5234134



登录 (/sign\_in) 发表评论



29条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



等风来whu (/u/05fcfc1c58df)

4楼 · 2016.12.20 13:05

(/u/05fcfc1c58df)

不错不错👍，不知道楼主愿意共享下代码吗？

👍 4人赞 🗨 回复



pennNnng (/u/16c8a625abb6)

2楼 · 2016.09.02 18:54

(/u/16c8a625abb6)

笔者写得不错

👍 赞 🗨 回复

一个不知死活的胖子 (/u/b04c0d6732ff)：@pennNnng (/users/16c8a625abb6) 嘿嘿

2016.09.03 09:22 🗨 回复

✍ 添加新评论



杨海宏 (/u/59c16657c292)

3楼 · 2016.11.03 17:17

(/u/59c16657c292)

你好，请问你能共享一份完整的代码吗？根据你的博文，我只能做到AUC=0.82, 另外，我发现携程的数据标注是有错误的。

👍 赞 🗨 回复

Slyne\_D (/u/8776bc99ada8)：我用了100维的词向量，SVM做了PCA之后的auc=0.88 acc=0.824  
MLP的auc=0.92 acc=0.854

2017.04.19 14:33 🗨 回复

Slyne\_D (/u/8776bc99ada8)：@Slyne\_D (/users/8776bc99ada8) 更新一下，还是100维的词向量作为Istm的预设置的embedding，最后的auc=0.94 acc=0.8847 5个epoch

2017.04.21 14:54 🗨 回复

冉工 (/u/a80e2580c6e9)：能参考下你的代码么。。我撸的

<http://www.aidnoob.com/suibi/2017/20170502/> (<http://www.aidnoob.com/suibi/2017/20170502/>)

感觉也是不对。

2017.05.02 15:14 🗨 回复

✍ 添加新评论 | 还有10条评论，展开查看



一个上进的胖子 (/u/62ffaec23bb7)

5楼 · 2017.02.28 21:22

(/u/62ffaec23bb7)

你好，可以发一份源码吗？

👍 赞 🗨 回复



JeremyF (/u/3f54b70c015b)

6楼 · 2017.03.16 00:02

(/u/3f54b70c015b)

你好，请问可以共享一份完整的源代码吗

👍 赞 🗨 回复



zhonglongshen (/u/0826c0d458b6)

7楼 · 2017.03.26 11:28

(/u/0826c0d458b6)

有代码吗

？



👍 赞    💬 回复



Radio\_Active (/u/16a8d68da88a)

8楼 · 2017.03.29 03:46

(/u/16a8d68da88a)  
可以分享一下代码吗？ 😊

👍 赞    💬 回复



大壳kk (/u/8c0ea39313dd)

9楼 · 2017.04.17 18:50

(/u/8c0ea39313dd)  
可以分享以下代码吗？

👍 赞    💬 回复



Tyroneme (/u/509f23e6b0a1)

10楼 · 2017.05.01 13:32

(/u/509f23e6b0a1)  
笔者您好 可以联系您吗

👍 赞    💬 回复



Tyroneme (/u/509f23e6b0a1)

11楼 · 2017.05.01 13:35

(/u/509f23e6b0a1)  
想请教您关于情感分析的事情，可以吗

👍 赞    💬 回复



冉工 (/u/a80e2580c6e9)

12楼 · 2017.05.02 15:13

(/u/a80e2580c6e9)  
<http://www.aidnoob.com/suibi/2017/20170502/>  
(<http://www.aidnoob.com/suibi/2017/20170502/>) 受人所托 撸了下 相关代码 但不知道 写得正确与否 忘指点一二。

👍 赞    💬 回复



写淡 (/u/6334438ddb5f)

13楼 · 2017.05.04 17:22

(/u/6334438ddb5f)  
请问，我已经下载所有的词库，发现文档里面没有分数，是自己加上去还是我下载方式不对？ 😊

👍 赞    💬 回复



镁博 (/u/5c9aa9261036)

14楼 · 2017.05.06 09:24

(/u/5c9aa9261036)  
赞赞！，待价而沽！，要完整的步骤设计和程序！

👍 赞    💬 回复



CakeMarmalade (/u/9d7f7a51c0a8)

15楼 · 2017.05.07 20:53

(/u/9d7f7a51c0a8)  
楼主大大，可以分享一下代码吗？ 😊 😊 😊

👍 赞    💬 回复





大小果果 (/u/0d6b479e438b)

16楼 · 2017.05.07 20:57

(/u/0d6b479e438b)

楼主不知能否共享一下代码？

赞 回复

被以下专题收入，发现更多相似内容

[算法 \(/c/f417a955c48c?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/f417a955c48c?utm_source=desktop&utm_medium=notes-included-collection)[python加油站 \(/c/1680257829b6?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/1680257829b6?utm_source=desktop&utm_medium=notes-included-collection)[文本挖掘 \(/c/459400e10ca3?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/459400e10ca3?utm_source=desktop&utm_medium=notes-included-collection)

更多精彩内容

### 办理国外学历学位认证（上海为例）2017/05 (/p/c95ecee7c246?

[utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation](/p/c95ecee7c246?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

拍拍屁股回国啦，发现回国后真的有好事情需要办理，我就把我的经历经验分享一下吧 0. 基于干啥先看先前经验的习惯，笔者本人主要参考的是上海国外学历学位认证总结 2016年12月，其他地方的具体情况请自...

[一个不知死活的胖子 \(/u/b04c0d6732ff?utm\\_campaign=maleskine&utm\\_content=user&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation\)](/u/b04c0d6732ff?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

### 问答机器人的Python分类器 (/p/8acb99b6671b?

[utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation](/p/8acb99b6671b?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

导语现在问答机器人真是火的不要不要的，大致分为两类：普适多场景的和单一专业场景的问答机器人。由于资源有限，不知死活的笔者只做了单一场景的分类器，如对海量数据、多场景的问答机器人感兴趣的话...

[一个不知死活的胖子 \(/u/b04c0d6732ff?utm\\_campaign=maleskine&utm\\_content=user&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation\)](/u/b04c0d6732ff?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

### 我删掉了游戏《王者荣耀》，同时删掉的还有今日头条 (/p/256b37a352a3?

[\(/p/256b37a352a3?utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation\)](/p/256b37a352a3?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)[utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation](/p/256b37a352a3?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

文 / 沐丞 最近身边很多人都在玩手游《王者荣耀》，有的人在聚餐等上菜的时候也不忘晚上一局，每天中午午休的时候也有不少同事都在拼命厮杀。有时候在咖啡馆，看电影等放映的时候都看到有很多人在玩。关...

[沐丞 \(/u/73fd48dcb7ba?utm\\_campaign=maleskine&utm\\_content=user&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation\)](/u/73fd48dcb7ba?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

### 高考失利、读二本院校的同学，后来都怎么样了？ (/p/e481f6df13eb?

[\(/p/e481f6df13eb?utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation\)](/p/e481f6df13eb?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)[utm\\_campaign=maleskine&utm\\_content=note&utm\\_medium=pc\\_all\\_hots&utm\\_source=recommendation](/p/e481f6df13eb?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

昨天晚上，高中同学微信群打破了久违的安静，不知道是不是受高考的影响，大家热火朝天的聊起了当年的高考，聊起了自己目前的工作生活。距离2010年高考，一晃七年过去了，不禁感叹时间流逝。 01 首先我...



樱花物語 (/u/ca34e04a051d?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

免费送你50本书，一起来画画吧！ (/p/7caf73c84d33? (/p/7caf73c84d33?  
utm\_campaign=maleskine&utm\_content=note&utm

utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)  
大家好，我是小荐荐。第三期送书写书评活动还没结束，我又来送书啦~ 这次简书荐书和漫画手绘专题联合  
上海人民美术出版社，将送出50本绘画教程类的书，想来想去，绘画类的书也不好写书评，所以这次采用“...  
小荐荐 (/u/28ee510656c6?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

