新闻分类问题

首先读取文件：

path1='C:/Users/Jaqen/Desktop/News Topic Classification/train\_texts.txt

'

path2='C:/Users/Jaqen/Desktop/News Topic Classification/train\_labels.tx

t'

path3='C:/Users/Jaqen/Desktop/News Topic Classification/test\_texts.txt'

path4='C:/Users/Jaqen/Desktop/News Topic Classification/test\_labels.txt

'

with open(path1) as f:

train\_texts=f.read().splitlines() #新闻内容训练集

with open(path2) as f:

train\_labels=f.read().splitlines() #新闻标题训练集

with open(path3) as f:

test\_texts=f.read().splitlines() #新闻内容测试集

with open(path4) as f:

test\_labels=f.read().splitlines() #新闻标题测试集

其中用splitlines（）将读取文件时每一行后的换行符去掉。我的方法是用Bag-of-words and its TFIDF提取特征，从sklearn.feature\_extraction.text导入TfidfVectorizer,因为这个库中有英文停用词，所以设置stopwords为’english’,为了防止内存错误，我取max\_feature为20000（原来去停用词再分词后总词数超过60000）。

#将文档数据转化为TF-IDF向量

tv=TfidfVectorizer(stop\_words='english',max\_features=20000)

tv.fit(train\_texts)

voca\_dict=tv.vocabulary\_

#print(voca\_dict) #打印词库

将训练集和测试集都转换成向量，其中x\_train为转换后的训练集矩阵，y\_train为训练集标签，x\_test为转换后的测试集矩阵，y\_test为测试集标签。

#数据向量化

x\_train=tv.fit\_transform(train\_texts).toarray()

y\_train=np.array(train\_labels)

x\_test=tv.transform(test\_texts).toarray()

y\_test=np.array(test\_labels)

定义一个评估函数erro\_rate(),即错误率，参数y\_predict为模型预测的标签向量，参数y为真实标签向量，计算公式为：预测错误的标签个数/标签总个数。

def error\_rate(y\_predict,y): #错误率

count=0

for i in range(y\_predict.shape[0]):

if y\_predict[i]!=y[i]:

count+=1

return count/y\_predict.shape[0]

从sklearn.navie\_bayes导入MultinomialNB,用朴素贝叶斯模型来进行分类工作。

#使用多项式分布的朴素贝叶斯算法训练

classifier=MultinomialNB()

classifier.fit(x\_train,y\_train)

y\_test\_predict=classifier.predict(x\_test) #测试集的预测值

y\_train\_predict=classifier.predict(x\_train) #训练集的预测值

使用sklearn中的classification\_report用来显示主要分类指标的文本报告，并打印出训练集和测试集的预测结果：

图片包含 户外, 文字

已生成高可信度的说明

可以看出训练集的错误率约为0.086092，测试集的错误率为0.0975，这个模型在测试集上的正确率达到了90.25%，在每个类上的精确度、召回率及F1值都在0.86以上，可见此模型在这个新闻分类任务上的效果还是不错的。

改进方法可能有用N-gram模型来进一步探索文本和新闻分类的关系，或对词库进行stemming或者lemmatization来缩减词库。