**阅读笔记**

**SnowNLP**是一个python写的类库，可以方便的处理中文文本内容，**中文分词**、**词性标注**、**情感分析**、**文本分类**、**提取文本关键词**、**文本相似度计算**等。它是受到了[TextBlob](https://github.com/sloria/TextBlob)的启发而写的，由于现在大部分的自然语言处理库基本都是针对英文的，于是写了一个方便处理中文的类库，并且和TextBlob不同的是，这里没有用NLTK，所有的算法都是自己实现的，并且自带了一些训练好的字典。

1.安装：pip3 install snownlp

2.基本用法：

from

snownlp import

SnowNLPs = SnowNLP(u'一次满意的购物')s.words

1) s.words 词语

2) s.sentences 句子

3) s.sentiments 情感偏向,0-1之间的浮点数，越靠近1越积极

4) s.pinyin 转为拼音

5) s.han 转为简体

6) s.keywords(n) 提取关键字,n默认为5

7) s.summary(n) 提取摘要,n默认为5

8) s.tf 计算term frequency词频

9) s.idf 计算inverse document frequency逆向文件频率

10) s.sim(doc,index) 计算相似度

这个sim的话，后面跟的是一个可以迭代的东西。

2.训练：

训练的文本是有要求的。

1)分词

比如说你要进行分词的话，首先先from snownlp import seg,然后通过seg.data\_path可以看到词典的路径。

我的路径是：

/usr/local/lib/python3.6/site-packages/snownlp/seg/seg.marshal

然后你可以看到在同一个目录下，有个data.txt，这个就是训练的样本，打开样本可以发现：

迈/b 向/e 充/b 满/e 希/b 望/e 的/s 新/s 世/b 纪/e

中/b 共/m 中/m 央/e 总/b 书/m 记/e

这样的句子

其中b代表begin，m代表middle，e代表end，分别代表一个词语的开始，中间词和结尾，s代表single，一个字是一个词的意思。

将训练样本放入data1.txt中，进行训练：

seg.train('data1.txt')

如果下次还需要使用的话，将结果保留到seg2.marshal当中，使用的时候只要更改data\_path改为seg2.marshal的路径即可。

2）情感分析

和上面一样，能看到/usr/local/lib/python3.6/site-packages/snownlp/sentiment里面有2个txt文件，neg.txt和pos.txt,这两个里放的是负向情感和正向情感的句子。所以在训练的时候也只需要这样子放入句子即可。

from snownlp import sentiment

sentiment.train(neg1.txt,pos1.txt)

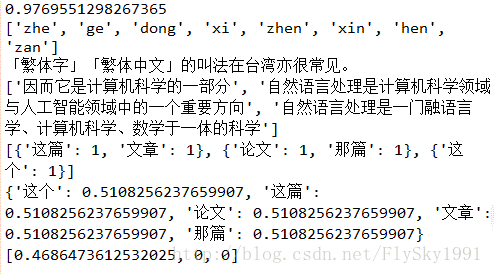
sentiment.save('sentiment2.marshal')

返回值为正面情绪的概率，   
越接近1表示正面情绪   
越接近0表示负面情绪

二、代码首页

分析

结果



我重点关注的：

前面除了刚才已经讲过的，我就不再重述了。

就讲一下我比较能懂的，已经感兴趣的部分。

评价词语对文本的重要程度

TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。

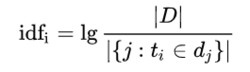
TF词频越大越重要，但是文中会的“的”，“你”等无意义词频很大，却信息量几乎为0，这种情况导致单纯看词频评价词语重要性是不准确的。因此加入了idf

IDF的主要思想是：如果包含词条t的文档越少，也就是n越小，IDF越大，则说明词条t越重要

TF-IDF综合起来，才能准确的综合的评价一词对文本的重要性。

TFIDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。TFIDF实际上是：TF \* IDF，TF词频(Term Frequency)，IDF逆向文件频率(Inverse Document Frequency)。TF表示词条在文档d中出现的频率。IDF的主要思想是：如果包含词条t的文档越少，也就是n越小，IDF越大，则说明词条t具有很好的类别区分能力。

**逆向文件频率**（inverse document frequency，IDF）是一个词语普遍重要性的度量。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取以10为底的[对数](https://baike.baidu.com/item/%E5%AF%B9%E6%95%B0)得到：



四、

因为我看网上很多人把它拿来做对购物体验的评论的情感分析。

然后我也有看到一些应用 我觉得有助于我们去理解这个库怎么用的 以及作用何在

下面就可以给大家举一个简单有趣的应用

（别人的代码）

是测试这个库对情感分析的准确率

看 13

分析

总结 不是很难 但是比较容易理解 自己也可以去试试

在有语料库的话

这个库还是准确定率比较高的

其他的代码有的也不太懂 就差不多了吧