一、情感分析名词概述

『What』

情感分析是文本分类的一个分支，是对带有情感色彩（褒义贬义/正向负向）的主观性文本进行分析，以确定该文本的观点、喜好、情感倾向。

例如说，文本"这是书读来爱不释手"归为正向，"这本书很难看"归为负向。当然也有层次更多的分类。

『Why』

被研究的主观性文本包括顾客对某个产品的评论，大众对某个新闻热点事件的观点等。通过这些文本，商家可以为消费者提供决策参考，相关机构也可以了解舆情，但人工分析耗费大量成本。

『How』

目前有基于情感词典的情感分析和基于机器学习的情感分析这两种主流方法。

【基于情感词典】是指根据已构建的情感词典，对待分析文本进行文本处理抽取情感词，计算该文本的情感倾向。最终分类效果取决于情感词典的完善性。

【基于机器学习】是指选取情感词作为特征词，将文本矩阵化，利用 logistic Regression, 朴素贝叶斯(Naive Bayes)，支持向量机（SVM）等方法进行分类。最终分类效果取决于训练文本的选择以及正确的情感标注。

（觉得有点抽象的后文有例解）

当然，特定情况下研究某些文本时也可以将两种方法结合起来。

比如说某些领域的文本没有标注，该领域的情感词典也不够完善，而人工标注需要耗费大量成本，数据的采集相对于人工成本小很多时；可以选取部分文本，利用基本情感词典的方法粗略地计算这些文本的情感得分值，选取分值偏高或偏低的文本作为已标注的训练文本 。再结合机器学习的方法进行分析。

（二）、情感分析流程例解

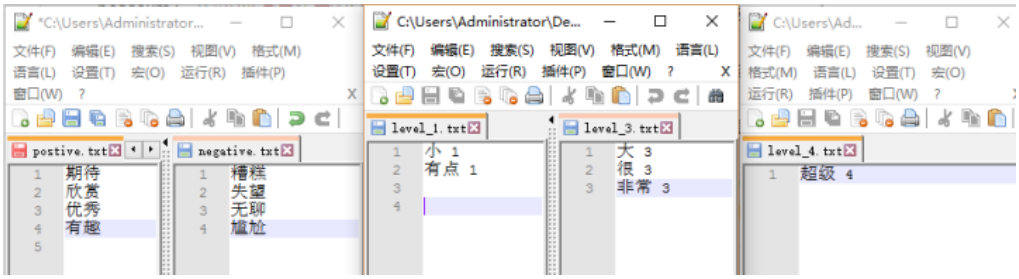
在本篇文章中，我试图用最简洁的语言和图例展现这两种分类方式。

（1）基于情感词典的情感分析

以下是我一条猫眼上选取的《西游伏妖篇》影评：

Review1 = “周星驰＋徐克，把观众期待度放到很大，看后小失望。特效场景、人物服饰、经典创新这些方面都很值得欣赏，可惜硬伤是一众主演演技尴尬，剧情超级无聊，走神好多次。”

我们现在有以下词典：



情感词：设定 positive 的词+1，negative 的词-1；

程度词：比如出现"小失望"就 -1*1，出现"非常失望"就-1*3

那么这句话的情感总分值就是 1*3-1*1+1*3-1-1*3 = 1

其中正向得分：1*3+1*3-1*3 = 3，负向得分 |-1*1-1|=2

可输出[review1 : 1] or [review1: (3,2)].

————————————————分界线————————————————

中文分词可利用 jieba, THULAC，ICTCLAS

该方法的重点在于『构建适合的情感词典』，不在此赘述，会在后续文中填上。

（2）基于机器学习的情感分析

若以下是一些政治方面的新闻文本

…	
第n条	XXXXXXXXXX
第n+1条	XXXXXXXXXX
第n+2条	XXXXXXXXXX
…	

1. 首先人工标注好其情感分类。正向为 1，负向为 0。
2. 我们要选取这些文本中的【特征词】，比如说做情感分类时的特征词要选取情感词，做商品分类时要选取商品名、商品特别的描述词等，构造词袋（bags of words）模型。
   即统计各词词频，形成如下词矩阵：

   其中 A,B,C,D... 代表各情感词，每一行代表一个文本，每个数代表在该文本中该特征词词频。

Classification	A	B	C	D	....
1	3	20	6	6	
0	10	12	2	4	
1	2	2	23	2	
0	5	3	18	4	
...					

   用 numpy 的数据类型将该数据储存起来

   【生成词向量的方法英文处理可考虑库 gensim，并且 Multiangle's Notepad 这篇博客有详细介绍“利用 gensim 和 sklearn 搭建一般文本分类器方法”】

   将数据集分为 n 份，其中 n-1 份为训练集，1 份测试集，从库 sklearn 中载入 svm, logisticRegression, NB 等分类算法。

```
train_set = data[1:i,:] #i-1 个数据作为训练集
test_set = data[i:,:] #剩余的数据作为测试集，训练集要远多于测试集

train = train_set[:,1:]
tag = train_set[:, 0] #第一列是类标签
import sklearn

from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB


clf_svm = svm.LinearSVC()
clf_svm_res = clf_svm.fit(train,tag)
train_pred  = clf_svm_res.predict(train)
test_pred   = clf_svm_res.predict(test_set)
```

```
clf_lr = LogisticRegression()
...

clf_nb = GaussianNB()
...
```

测试集轮换重复 n 次，使用交叉验证测试分类器的准确度。在此另 n=10.
比较不同分类器的准确率，比较选出最优的。

```
from sklearn import cross_validation

kfold = cross_validation.KFold(len(x1), n_folds=10)

svc_accuracy = cross_validation.cross_val_score(clf_svm, train, tag, cv=kfold)


print 'SVM average accuary: %f' %svc_accuracy.mean()

...
```

————————————分界线————————————

这种方法的重点在于特征选取。

来源于 https://zhuanlan.zhihu.com/p/25065579