情感极性分析

班级：2016211303　　姓名：黄若鹏　　学号：2016212901

1.题目

- Detecting sentiment polarity: 40 points
  -Given text about movie reviews
  -Can we detect sentiment,like whether a comment is
  * Positive?
  * Negative?
  -Can we tell to what extent is a comment positive of negative?
- Data
  -5331 positive snippets
  -5331 negative snippets

- Other resources:
  -The Subjectivity Lexicon

2.概要

使用朴素贝叶斯分类算法

朴素贝叶斯分类算法，是一种有监督学习算法，通过对训练集的学习，基于先验概率与贝叶斯公式，计算出特定条件下样本属于某一类别的概率(条件概率)，从而达到分类的目的。

假设词和词在表达句子的意思时互相独立的，根据贝叶斯公式

1、找到一个已知分类的待分类项集合，这个集合叫做训练样本集。

2、统计得到在各类别下各个特征属性的条件概率估计。即
$P(a_1|y_1), P(a_2|y_1), ..., P(a_m|y_1); P(a_1|y_2), P(a_2|y_2), ..., P(a_m|y_2); ...; P(a_1|y_n), P(a_2|y_n), ..., P(a_m|y_n)$
。

3、如果各个特征属性是条件独立的，则根据贝叶斯定理有如下推导：

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

- 先选出前 80%作为训练集，后 20%为测试集。

- 在训练集中，统计所有单词正面和负面情绪词出现的比例，然后在做归一化处理。
- 在测试集中，先分割单词，对于出现在词性单词集中的单词，判断是属于哪一类单词，是属于正面词性还是属于负面词性。最后遇到语句终结符句号时，计算正因子和负因子之间的距离，判断该句是正面评价，还是负面评价。
- 这里很难处理的就是正话反说，所以难免会误判，导致，很多句子判断出错，比如 *Who does not know*？ 这个句子翻译过来是人人都知道，应该属于正面句子，但是由于这个句子是正话反说，结果却被认定为负面情绪。 可以分析得知，这种句子以问句居多， 因此为了改进，可以考虑对待以问号结尾的句子，加上调整比例因子，使得，被误判的可能性降低。
- 这里我在网上看到各种博客，其中有考虑去除训练集中的停止词，这样可以使得，训练出来的效果，不会受到这些词的影响（停止词，中文中，呢，了之类的；英文中，the，a 之类的），
- 最后在判定啊结果的时候，计算准确率和召回率，最后计算 F-score

$$\text{precision} = \frac{true\ positive}{true\ positive + false\ positive} = \frac{true\ positive}{no.of\ predicted\ positive}$$

$$\text{recall} = \frac{true\ positive}{true\ positive + false\ negative} = \frac{true\ positive}{no.of\ actual\ positive}$$

$$\text{F1Score} = 2 \times \frac{precision \times recall}{precision + recall}$$

3.运行环境

系统：windows10

软件：VsCode

4.输入输出

输入：已标分好的 positive 词汇，和 negative。 这里.NEG 和.POS 并不代表文件类型，只代表数据类型，打开时需要用 UTF-8 解码

输出：负面词和，正面词的准确度以及 F-score

5.结果分析

```
PS C:\Users\Rocair\Desktop\finalNLP\bayes情感分析> cd
HONIOENCODING}='UTF-8'; ${env:PYTHONUNBUFFERED}='1';
hon-2018.12.1\pythonFiles\ptvsd_launcher.py' '--defau
rs\Rocair\Desktop\finalNLP\bayes情感分析\SentimentAna
Pos Precision: 0.7742520398912058
Pos Recall: 0.801125703564728
Pos F-score: 0.7874596588289536
Neg Precision: 0.7941747572815534
Neg Recall: 0.7666354264292409
Neg F-score: 0.7801621363853124
```

后来我把训练集和测试集的比例中心调整了一下，发现准确度提高并不明显，

```
PS C:\Users\Rocair\Desktop\finalNLP\bayes情感分析> cd
HONIOENCODING}='UTF-8'; ${env:PYTHONUNBUFFERED}='1';
hon-2018.12.1\pythonFiles\ptvsd_launcher.py' '--defau
rs\Rocair\Desktop\finalNLP\bayes情感分析\SentimentAna
Pos Precision: 0.7822111388196176
Pos Recall: 0.8022165387894288
Pos F-score: 0.792087542087542
Neg Precision: 0.7972027972027972
Neg Recall: 0.7768313458262351
Neg F-score: 0.7868852459016394
```

分析原因可能是和数据集有很大的关系，由于数据集的数量很大，训练语料的有效分布比较均匀，因此出现上述情况。

6.实验部分代码

```
    def Bayes(self):
        for line in self.temp_pos_test:#判断正面词
            pos_factor = 1.0
            neg_factor = 1.0
            for word in line.split():
                if word in self.subj:
                    pos_factor*= self.Pos_Data[word]
                    neg_factor*= self.Neg_Data[word]
            if pos_factor >= neg_factor:
                tp +=1
            else:
                fn +=1

        for line in self.temp_neg_test:#判断消极词
            pos_factor = 1.0
            neg_factor = 1.0
            for word in line.split():
                if word in self.subj:
                    pos_factor *= self.Pos_Data[word]
                    neg_factor *= self.Neg_Data[word]
            if pos_factor > neg_factor:
                fp+=1
            else:
                tn+=1
```