**基于朴素Bayes的情感分类器 设计文档**

1. **任务定义**

在给定情感词词典（包括情感词的程度强弱、极性、词性等信息）和带情感标记的正负情感句子集的情况下，训练一个分类器，这个分类器可以对输入的句子进行情感极性判断，即将其归类为正面或负面情感。在本实验中，共进行了如下几个方面的尝试：使用否定词词典扩充特征，使用基于情感词典的统计分类，使用基于朴素Bayes的情感分类，使用上述两种方法相结合的混合分类。

1. **源码运行环境**

采用python2.7编程，涉及到numpy的一些基本操作，编程环境为win10操作系统下的wing IDE 5.1 Pro。

1. **输入输出**

**程序输入：**

本程序涉及两个输入集，一个是情感文本，一个是情感词典。

其中情感文本来源于rt-polarity.pos和rt-polarity.neg，前者包含了5331条关于电影评论的正面评价，后者包含了相同数量的负面评价。将情感文本中4000条正文本和4000条负文本作为训练集，其余数据均作为测试集。

情感词典来源于subjclueslen1-HLTEMNLP05.tff，其内容是格式化的，包括了一个情感词本身的词性、是否为词干、情感强度、情感极性等信息，在本次实验中，用到了情感词的情感强度和情感极性这两个信息。

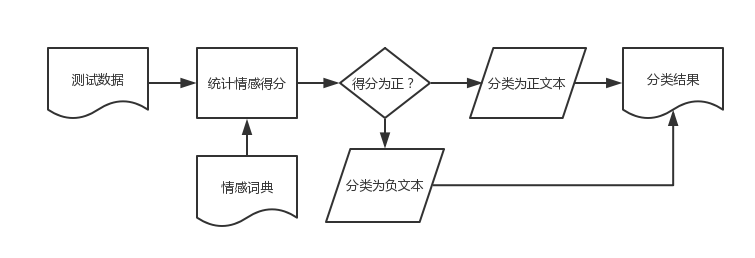
**程序输出：**

出于简洁的目的，本实验不再专门地以文件形式输出情感分类结果，而是以正确率的形式直接在命令行输出一些分类方法的准确率，使各类模型的性能一目了然。

1. **方法描述**
2. **实验总体描述**

本实验还原了刚刚接触情感分类的一些方法的尝试，包括了纯基于情感词典的打分式分类，得知效率过查后考虑机器学习的分类方法，包括基于朴素Bayes的二分类器。在得知否定词的存在可能影响分类性能后，我们进一步为分类器添加否定词词典，与原有的情感词典共同作为提取特征的手段。最后，我们试图将二者结合起来，即先将一部分测试数据先通过基于情感词典的方式筛选出来，将情感得分较为模糊的剩余测试数据置入已训练好的Bayes分类器中，共同得出最后的分类结果。

1. **基于情感词典的分类器**

此类方法十分简单，可用如下流程图表述。

可以看出，情感的判断完全基于情感词的统计结果。在情感词典中，我们规定积极情感词如果为强类型，则得分为+2，如果为弱类型，则得分为+1；同理，消极情感词为-2和-1。对于中性情感词，得分为0。

要做的事很简单，将每一条评论切分成单词之后，可以用如下伪代码统计一个句子的情感得分。

*def 评价一个句子的情感得分：*

*情感得分 = 0*

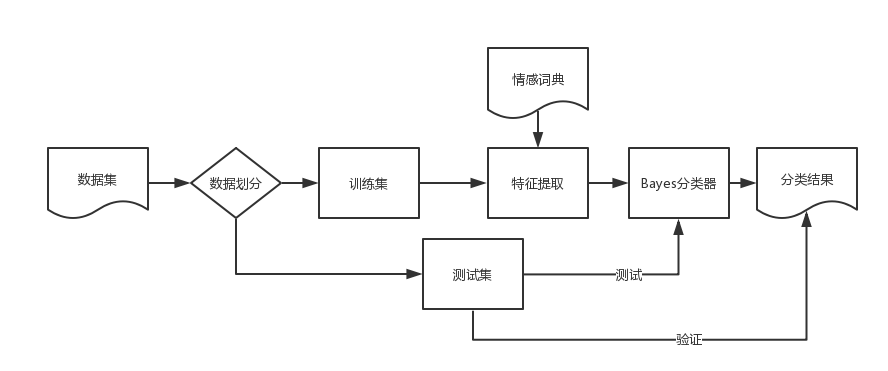
*for 句子中每一个词：*

*if 这个词在情感词典中：*

*情感得分 += 这个词的情感得分*

*return 情感得分*

1. **基于朴素Bayes的分类器**

基本的流程可以表述如下：

1. **数据预处理**

同HMM词性标注中叙述的类似，我们借助python的re模块完成对情感词典中的所有词的提取。同时，将数据集切割为训练集和测试集，并将每一条句子抽象化为python的list，同时去除标点符号。

1. **特征提取**

我们选择使用了词向量表示法来表示一个文本（一条评论），其中向量的维数为情感词典的宽度，也就是说，我们选择情感词典中的每一个词作为一条评论的特征。

在特征加权方面，我们选择对特征进行bool加权，这是出于以下的考虑：其一，不同于普通的文本分类，一个情感词在其他文档中的频次是不必要的特征，因此我们舍弃关于IDF的考虑；其二，由于数据均为较短的评论，因此在TF特征中采用bool加权和直接统计频次的效果差异不大；其三在Pang和Lee的论文中观察到，最优结果是在使用bool值和unigram相结合的方式得到，而非绝对频次和unigram相结合，因此我们最终选用bool加权的方式提取文本的特征。

至此，一个文本得以以向量的形式表示，便于我们之后的训练和分类。

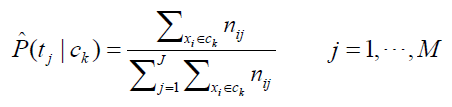
1. **构建朴素Bayes分类器**

朴素Bayes分类器基于这样一个假设，即如果分类C是确定的，那么变量X1，X2…Xn之间相互独立，这为缩小参数空间提供了可能，即

如果给定一个xi，求解其最可能的类别，这个问题的求解可以被描述如下：

采用log的形式是为了防止数据下溢，我们只需对上式的一些参数做一些讨论。

第一个log项可以从给定的训练语料中进行统计，在本实验中，由于正负训练文本数量相同，显然P（neg） = P（pos） = 0.5

第二个log项的每一个加项都可以用MLE估计，可以表述为

凡是用到MLE的地方都不能忽略数据稀疏问题，与之前的任务类似，我们采用简单的delta=0.5的平滑方法来平滑MLE的值。

至此，对于xi，我们已经可以算出其在每一个分类的预测值，Bayes分类器的分类过程就是将xi归为预测值最大的那一类。在本实验中，这种分类问题退化为二分类问题。

训练Bayes的过程在python中十分简单，可采用如下伪代码实现：

*初始化向量v，其维度为情感词典的宽度*

*for 训练集中的每一条评论：*

*for 评论中的每一个词：*

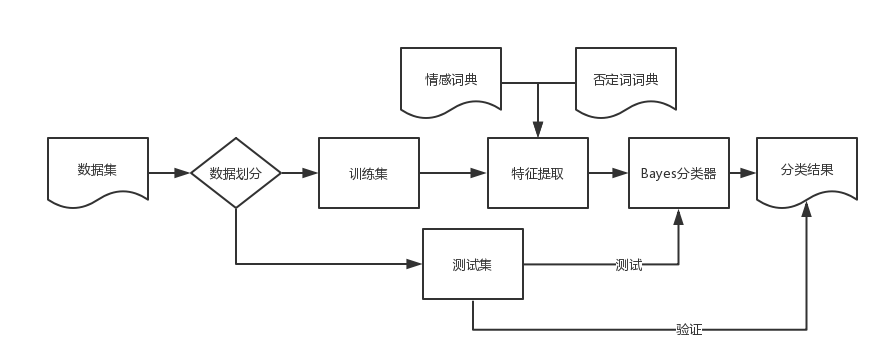
*if 它在情感词典中：*

*v[这个词对应的索引] += 1*

*for 向量v中的每一个元素：*

*计算MLE估计值，同时采用delta=0.5的平滑算法*

此时我们得到了三个返回值，pos\_vec代表了分类器中的积极类判断向量、neg\_vec代表了消极类别判断向量、p\_pos是训练集中pos文本出现的概率。由于是二分类问题，显然p\_neg = 1 – p\_pos。

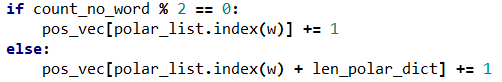
1. **否定词特征的引入**

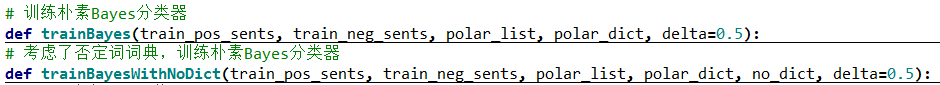
我们仍需考虑一个问题，英文中类似not这类的否定词还有should这样的假设词会使其后跟随的情感词的极性取反，也就是说，这类的否定词也应该纳入考虑中。因此我们继续改造我们特征提取的过程，表述如下：

我们如果出现了上述的情况，如happy前跟了一个not，那么我们为这个文本添加新的否定特征，即NOT\_HAPPY。对于情感词典中的每一个词都做这样的处理，此时特征维数是原来的两倍。

再进一步考虑，否定词在一个情感词之前可能不止一次出现，因此我们需要引入滑动窗口的机制来统计一个词之前的否定成分，在否定词生效过后，我们应该及时清理这个统计，使其不会对之后的情感词造成影响。

上述过程由mixCLF中的polarMarker过程体现，具体实现十分简单，不再赘述。同时，我们也要改变Bayes分类器中的向量。

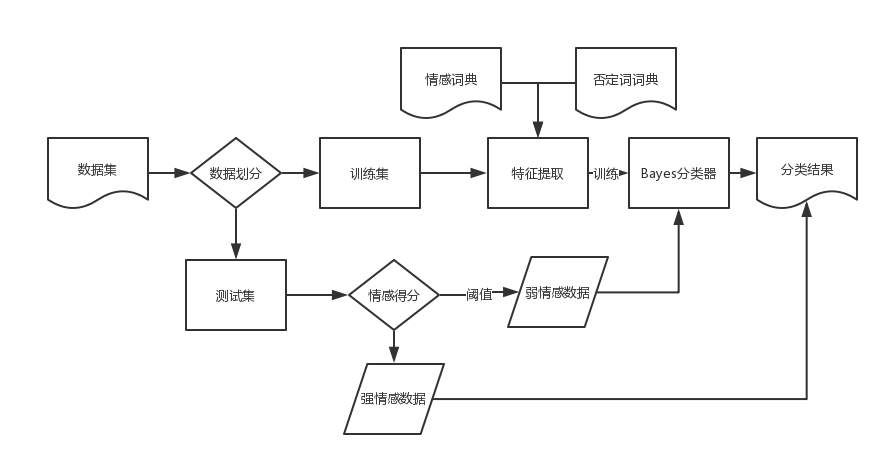
同时在特征提取时增加对否定词个数的判断。

它与原来的差别只是增加了对否定词词典的考虑。

在实验结果中表明，这样的考虑会带来准确率方面的提升。

1. **基于情感词典和朴素Bayes的混合分类器**

是否能将以上两种方法结合？我们知道基于情感词典的方法是粗糙的，但是如果一个句子的情感得分超过一定的阈值，我们可以很确信的判断它的情感倾向。而对于那些情感得分较为模糊的句子，我们再用Bayes分类器进行处理，也许会得到更好的效果。

这个过程进一步用图表示如下：

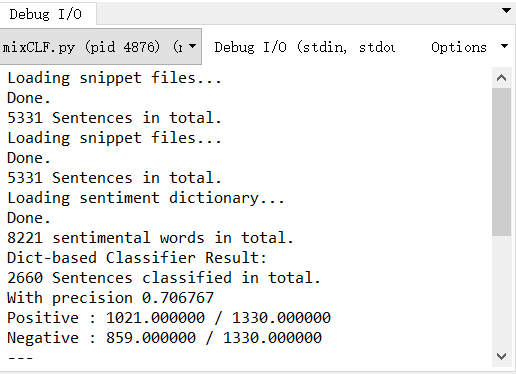
先将测试集中的所有评论计算情感得分，排序，根据一个阈值选取那些情感倾向最强的一些数据，直接确定它们的分类。对于剩余的情感模糊的数据，将它们放入训练好的Bayes分类器中进行再分类。

1. **结果分析及性能评价**

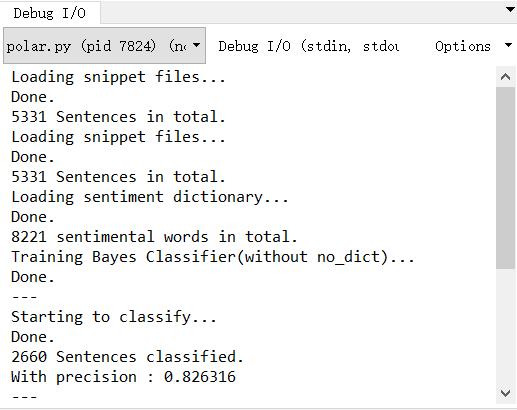
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **分类方法** | **情感词典** | **朴素Bayes（无否定词特征）** | **朴素Bayes（否定词特征）** | **混合分类器** |
| **准确率** | 70.6% | 82.6% | 83.9% | 83.6% |

1. **基于情感词典的分类结果**

通过命令行对准确率进行输出。

不难发现这样简单的方法分类效果只有约70%，这不是一个好的分类结果。

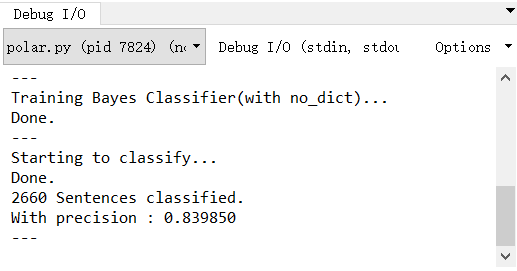
1. **基于朴素Bayes的分类结果（不考虑否定词）**

在不考虑否定词的情况下，我们对相同的测试集进行分类，结果如下：

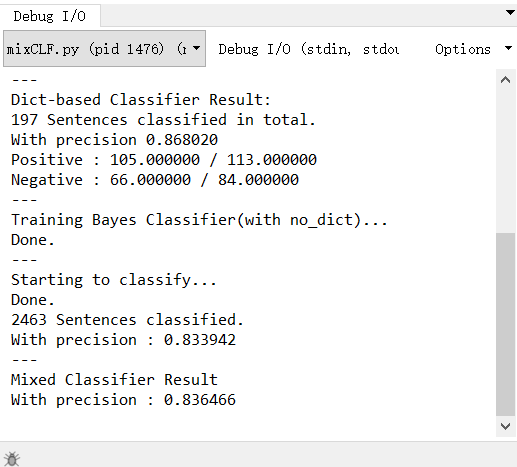
准确率已经提高到82%，整个分类过程约10s，这说明基于朴素Bayes分类的方法要远好于仅仅基于情感词典的统计方法。

1. **基于朴素Bayes的分类结果（考虑否定词）**

添加否定词词典作为新特征后，对相同的测试集进行分类，结果如下：

考虑了否定词后，分类器性能约提升了1.5%，分类时间几乎相同。

1. **基于混合分类器的分类结果**

在混合了两种方法后，测试结果如下：

这里我们需要详细解释一下各个准确率的含义。

先基于情感词典进行分类，由于阈值的限定，第一分类器只挑选了197条评论进行分类，原因是它们具有较高的情感得分，如此分类的准确率为86.8%，超过了朴素Bayes的准确率（这是因为较高的情感得分往往显示了很强的倾向性），随后，将剩余的2463个模糊评论交由Bayes分类器处理，在这个阶段，Bayes分类器的准确率是83.4%，总体来看，整个过程的总体准确率为83.6%，这要略差于完全基于Bayes的分类。

这是为什么呢？简单分析来看，分离出的“强情感倾向”的数据确实准确率较高，但获得这些较高准确率分类的前提是一定的阈值（情感有多强才算高），阈值越高造成被有效分类的数据越少，但降低阈值会降低第一次分类的准确率（阈值=0时退化成基于情感词典的统计方法，这种方法并不可靠）。因此总体而言，这种方法带来的改进效果是微乎其微的。