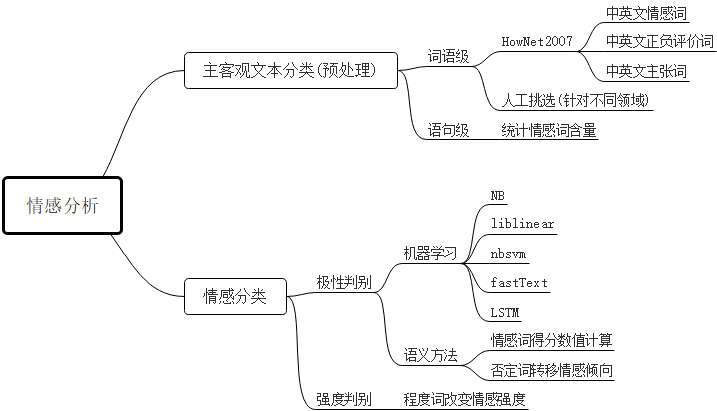
# 短文本情感分析概述

情感分析的应用场景很多，下边介绍几种。第1种电子商务方面，利用文本情感分析技术对产品评论观点进行组织和分类。第2种社会舆情分析方面，利用文本情感分析技术，可以更加及时地了解**网络民意**，使民间智慧与官方智慧良性互动。第3种影视评价方面，文本情感分析技术可以实现影评的自动分类，有利于用户快速浏览正反两方面的评论意见，减少观看影视时的盲目性。 第4种酒店服务评价。酒店可以根据用户的评论，了解他们的现实需求，根据需求提升酒店在相关问题上的服务质量。情感分析分为以下三个分支任务包括**主客观文本分类**、**情感极性判别**和**情感强度判别**。情感分析的任务可以用左图表示：第1部分主客观文本分类，它的主要任务是判断一个句子是否带有主观评价色彩。往往这样的句子中会含有特定类别的词语，因此可以从词语级进行判断。这些词语大致可以分为三类，第1类就是中英文**情感词**，第2类是正负**评价词**，第3类是中英文**主张词**。最后对于一句话，通过统计情感词含量来判断它是否属于主观文本。至此，第1部分的工作结束，这1部分也可以看作是系统的预处理过程，也就是说当输入的文本是客观表述时，不需要后续的处理，直接认为它的情感是中性的。第2部分就是情感分类了。在这里主要使用机器学习方法，对一句话进行正负类别预测，也就是极性判别。常见的适用于情感分类的机器学习方法有**朴素贝叶斯**，线性SVM，nbsvm，fastText，LSTM。另外一种针对情感分类的方法就是通过语义层面判别，主要是统计情感词数量。除了极性判别以外，还有情感强度计算，也就是通过**强度词**来衡量一个句子的情感强度。

# 情感分析语料和词典资源

在英文词汇资源方面，主要有General Inquirer（GI）和WordNet，其中GI词典（1966年开发）是英文文本情感分析研究中常用的基础资源之一。该词典包含182个词语类目以及11788个英语词汇。其中，有1915个词汇标注了“褒义”属性，2291个词语标注了“贬义”属性。对于一个词汇的多个义项，词典中将之作为不同条目列出，用于区分某个词语在特定义项或词性上体现的不同褒贬属性。GI的构建依赖手工标注，这些手工标注的词语情感倾向信息成为许多相关实验和研究的基础。

可用于情感词词表的构建的中文资源有以下几种：

（1）知网（HowNet）知网是一个以汉语和英语所代表的概念为描述对象，以表示概念与概念之间，以及概念所具有的属性之间的关系为基本内容的常识知识库。知网2007版情感分析用词语集提供了正面评价词3730个，负面评价词3116个，正面情感词836个，负面情感词1254个。这里的评价词是指消费者对产品及其特征发表的“肯定”或“否定”的评价，情感词是指对评论涉及的主题进行各种情感表达。

（2）《学生褒贬义词典》（张伟，2004）该词典收录了含有褒义或贬义的双音词、成语和惯用语共1672条，其中，褒义词728条，贬义词942条，兼具褒贬义的词2条。其中形容词446条，名词257条，动词393条。该词典的优点在于每个词条不仅标注了词性和褒贬色彩（褒义和贬义），还为词条标注了感情特点（赞扬、喜爱、批评、厌恶）和近义词等信息。

（3）《褒义词词典》（史继林，2005）。该词典共收录了5067个词条。

（4）《贬义词词典》（杨玲，2005）。该词典共收录了3495个词条。

获取词典资源，用于主客观评论分类

<https://download.csdn.net/download/kjp330/10106176>

获取语料

|  |  |
| --- | --- |
| 新浪微博开放平台 | <http://open.weibo.com/?sudaref=www.baidu.com&display=0&retcode=6102> |
| SemEval-2016 | <http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools> |
| 500万条微博语料 | <https://blog.csdn.net/karamos/article/details/80132231> |
| 中文情感分析语料整理 | <https://download.csdn.net/download/xhyqlbd/10600730> |

**资源小结**

**语料资源** <https://github.com/baiziyuandyufei/sentiment_analysis/tree/master/data/evaluation_corpus>

* COAE2014评测语料

task1 长新闻文本中抽取主观句并判断极性。语料未经过标注。

task2 非中文多语种文本极性判别。对中文文本情感分析没有太大用处。

task3 蒙牛评论语料，给定旧词典，要求抽取出新的词语并判断极性。语料极性已标注。领域太窄，没有太大用处。

task4 微博短文本极性判别，正、负、中。语料未标注极性

task5 微博短文本抽取评价对象（实体识别）语料未标注极性。

* COAE2015评测语料

微博短文本极性判别，正、负、中。语料已标注极性。

* 谭松波酒店评论语料

酒店评论语料，可以作为训练语料，已标注。

* 微博评论语料

微博短文本语料，可以作为训练语料，已标注。

**词典资源**

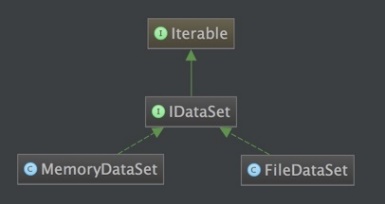
<https://github.com/baiziyuandyufei/sentiment_analysis/tree/master/data/collected_dictionay>

* 普通情感词汇词典资源
* 情感词汇本体词典资源

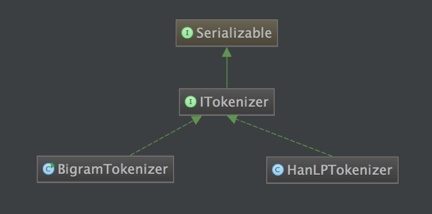
# PyHanLP搭建短文本情感分析系统

## HanLP中有关情感分析的三个类

### 数据加载存储类



1. MemoryDataSet是将训练数据全部加载到内存的，FileDataSet是利用文件系统存储数据的。
2. 两个类MemoryDataSet和FileDataSet都继承自AbstractDataSet类，这个类中有一个tokenizer实例，它是某一种分词方法实例



这里有两个具体分词类可供使用，其中BigramTokenizer是考虑的2grams字串作为一个词语，HanLPTokenizer使用的是实词分词器。

### 模型存储类

这里的类结构是有一个AbstractModel，NaiveBayesModel实现了该抽象类。类中主要存储模型中的样本数量，类别数量，特征数量，特征词语，分词器，类目名称映射，针对朴素贝叶斯模型还有似然概率以及先验概率。

### 贝叶斯分类器类

HanLP中有一个分类器接口IClassifier，抽象类AbstractClassifier实现了该接口，所有具体的分类器都继承自抽象类AbstractClassifier，提供独有的训练和预测方法。常用的预测方法为classify给出最可能的类目，predict给出属于每个类目的可能性。

## 生成目录-文件夹结构的训练数据

HanLP可以自动加载按照类目体系存储的语料文件。对于本系统来说，我们将微博语料的正、负极性语料文件存储为下边目录结构。

<训练语料文件的目录结构>

corpus

--neg60000.txt

--pos60000.txt

|  |
| --- |
| #coding:utf-8  """  训练语料按类目存储在不同文件中，将数据拆分成目录结构  <训练语料文件的目录结构>  corpus  --neg60000.txt  --pos60000.txt  注意：  1. corpus根目录名称任意  2. 两个类目文件neg60000.txt和pos60000.txt文件名任意但是必须以.txt结尾  3. 训练文件的编码由参数file\_coding指定，输出的文件都为utf-8编码  4. 训练文件中每行被认为是一个文本  5. 训练文件名被认为是目录名称（类目名称）  6. 支持多个类目文件处理（corpus目录中除了.py文件外任何文件都被认为是一个类目的训练文件）  (请将此脚本文件与待处理语料文件放在相同目录下)  """  import codecs  import os  # 递归删除某一目录下的所有文件，包括子目录下的文件  def del\_file(path):  ls = os.listdir(path)  for i in ls:  c\_path = os.path.join(path, i)  if os.path.isdir(c\_path):  del\_file(c\_path)  else:  os.remove(c\_path)  # 列出目录下的训练语料文件  def list\_files(path):  files\_li = []  for root, dirs, files in os.walk(path):  for one\_file in files:  if os.path.getsize(u"%s/%s" % (root, one\_file)) > 0 and one\_file.find(".py") == -1:  files\_li.append(u"%s/%s" % (root, one\_file))  return files\_li  if \_\_name\_\_ == '\_\_main\_\_':  # 训练文件的文件编码  file\_coding = 'gbk'  # 创建与清空train目录  if not os.path.exists('train'):  os.mkdir('train')  del\_file('train')  # 列出类目文件列表  files\_li = list\_files(".")  # 生成训练数据  for file\_name in files\_li:  label\_name = file\_name.strip("./").strip(".txt")  print("file\_name=", file\_name, "label\_name=", label\_name)  os.mkdir(u'train/%s' % label\_name)  file\_cnt = 0  with codecs.open(file\_name, 'rb', file\_coding, 'ignore') as infile:  for line in infile:  line = line.strip()  if line:  with open('train/%s/%d.txt' % (label\_name, file\_cnt), 'wb') as outfile:  out\_str = u'%s\n' % line  outfile.write(out\_str.encode('utf-8', 'ignore'))  file\_cnt += 1 |

## 训练朴素贝叶斯分类器

首先将得到的训练数据放在合适的位置，对于windows+anaconda的话，可以将数据放在C:\ProgramData\Anaconda3\Lib\site-packages\pyhanlp\static\data\test目录下。加载/训练朴素贝叶斯分类器的代码如下

|  |
| --- |
| **def** train\_or\_load\_classifier():  *# 朴素贝叶斯模型文件名* model\_path = sentiment\_corpus\_path + **'.ser'** *# 检查模型文件是否存在，如果存在则加载模型并返回朴素贝叶斯分类器对象* **if** os.path.isfile(model\_path):  **return** NaiveBayesClassifier(IOUtil.readObjectFrom(model\_path))  *# 模型文件不存在，则首先构建朴素贝叶斯分类器实例* classifier = NaiveBayesClassifier()  *# 传入训练文件路径名称进行训练* classifier.train(sentiment\_corpus\_path)  *# 获取训练后得到的模型* model = classifier.getModel()  *# 保存模型为模型文件* IOUtil.saveObjectTo(model, model\_path)  *# 传入模型到朴素贝叶斯分类器并返回朴素贝叶斯分类器对象* **return** NaiveBayesClassifier(model) |

## 朴素贝叶斯分类器预测

### 直接预测

|  |
| --- |
| **def** predict(classifier, text):  print(**"《%16s》\t属于分类\t【%s】"** % (text, classifier.classify(text)))  *# 如需获取离散型随机变量的分布，请使用predict接口  # print("《%16s》\t属于分类\t【%s】" % (text, classifier.predict(text)))* |

### 预测前首先进行主、客观分类

# 实践日志

|  |  |
| --- | --- |
| 20190419 | 1. 情感分析评测训练语料整理 2. 情感分析词典资源整理 |
| 20190422 | 1. 将新浪微博12万语料作为正、负极性训练集，训练朴素贝叶斯分类器 2. 直接预测最合适情感极性 |