情感分析与计算-课程报告

周牧云
1180300315

摘要：本次实验为情感三分类问题。训练数据为 8606 条，测试数据共 3000 条。使用机器学习的 SVM 分类方法，最终 实验正确率为 69%左右。

一、情感分析综述

情感分析也称为意见挖掘，是自然语言处理（NLP）中的一个领域，它试图在文本中识别和提取意见。

情感分析有很多的应用场景，例如社交媒体监控、品牌监控、客户之声、客户服务、员工分析、产品分析、市场研究与分析等等。

实现情感分析的方法有很多，大体上分为两大类，第一类为基于词典规则的方法，第二类为基于机器学习的方法。

1、基于词典的方法

基于词典的方法主要通过制定一系列的情感词典和规则，对文本进行拆句、分析及匹配词典（一般有词性分析，句法依存分析），计算情感值，最后通过情感值来作为文本的情感倾向判断的依据。

基于词典的情感分析大致步骤如下：

1）对大于句子力度的文本进行拆解句子操作，以句子为最小分析单元；

2）分析句子中出现的词语并按照情感词典匹配；

3）处理否定逻辑及转折逻辑；

4）计算整句情感词得分（根据词语不同，极性不同，程度不同等因素进行加权求和）；

5）根据情感得分输出句子情感倾向性。

如果是对篇章或者段落级别的情感分析任务，按照具体的情况，可以以对每个句子进行单一情感分析并融合的形式进行，也可以先抽取情感主题句后进行句子情感分析，得到最终情感分析结果。

2、基于机器学习的方法

机器学习的方法是将情感分析作为一个有监督的分类问题。对于情感极性的判断，将目标情感分为三类：正、中、负。对训练文本进行人工标注，然后进行有监督的机器学习过程，并对测试数据用模型来预测结果。

基于机器学习的情感分析大致步骤如下：

首先进行文本预处理。文本的预处理过程是使用机器学习作用于文本分类的基础操作。由于文本是非结构化数据及其特殊性，计算机并不能直接理解，所以需要一系列的预处理操作后，转换为计算机可以处理的结构化数据。在实际分析中，

文本更为复杂，书写规范也更为随意，且很有可能掺杂部分噪声数据。整体上来说，文本预处理模块包括去噪、特征提取、文本结构化表示等。

1）特征抽取：

中文最小语素是字，但是往往词语才具有更明确的语义信息，但是随着分词，可能出现词语关系丢失的情况。n-元文法正好解决了这个问题，它也是传统机器学习分类任务中最常用的方法。

2）文本向量化：

对抽取出来的特征，向量化是一个很重要的过程，是实现由人可以理解的文本转换为计算机可以处理数据的重要一步。这一步最常用到的就是词袋模型（bag-of-words ）以及最近新出的连续分布词向量模型（word Embedding）。词袋模型长度为整个词表的长度，词语对应维度置为词频，文档的表示往往比较稀疏且维度较高。Embedding 的表示方式，能够有效的解决数据稀疏且降维到固定维度，更好的表示语义信息。对于文档表示，词袋模型可以直接叠加，而 Embedding 的方法可以使用深度学习的方法，通过 pooling 得到最终表示。

3）特征选择：

在机器学习分类算法的使用过程中，特征好坏直接影响机器的准确率及召回率。选择有利于分类的特征，可以有效的减少训练开支及防止模型过拟合，尤其是数据量较大的情况下，这一部分工作的重要性更加明显。其选择方法为，将所有的训练语料输入，通过一定的方法，选择最有效的特征，主要的方法有卡方，信息熵，dp 深层感知器等等。

目前也有一些方法，从比句子粒度更细的层次去识别情感，如基于方面的情感分析（Aspect based Sentiment Analysis），他们从产品的评价属性等更细粒度的方面对评价主体进行情感倾向性分析。

文本转换为机器可处理的结构后，接下来便要选择进行机器学习的分类算法。目前，使用率比较高的是深度学习（CNN，RNN）和支持向量机（SVM）。深度学习的方法，运算量大，准确率有一定的提高，所以都在做这方面的尝试。而支持向量机则是比较传统的方法，其准确率及数据处理能力也比较出色，很多人都在用它来做分类任务。

二、系统方法

本次实验的任务如下：

设计和实现分类系统，完成对文本的情感分类任务，这里包含三种情感：中性，积极和消极。程序语言、框架、学习方法不限，可使用外部语料，不可使用已有的情感分析或文本分类库。

由于不能使用现有的词典，考虑到仅仅用现有的 8606 条样例可能无法搭建出一个精准度较高的词典，如果使用基于词典的方法可能准确率不尽如人意。同时，基于词典的方法的召回率一般而言会比较低。因此，我优先准备使用机器学习的方法，将情感分析当作一个有监督的三分类问题。分类算法方面，我是用支持向量机（SVM），主要是因为 SVM 分类速度比较快，也比较容易实现，而且准确率也有一定的保证。（还因为不会 LSTM）

三、实验设置

实验使用机器学习方法，其中分词工具使用 jieba 分词，使用 sklearn 中的 TfidfVectorizer 来进行词组向量化，使用 sklearn 中的 MultinomialNB 来进行标签分类。

代码说明如下：

```
def read_json(path):
    f = open(path, 'rb')
    return json.load(f)
```

该函数用于读取 json 类型的数据，采用二进制读入，避免了由于中文产生的问题。

```
def analyzes(text):
```

该函数用于处理读入的数据，首先进行分词操作，如果数据带标签，那么就给分好的词附带标签，以便之后进行分类。具体核心实现如下：

```
for line in text:
    t = re.sub('[’ @!,.:;\'\"#! ？。:;、]+', '', line['content'])
    words.append(' '.join(jieba.lcut(t, cut_all=True)))
```

读入的数据是一个字典,键分别为‘id’、‘content’、‘label’,其中‘content’索引到的内容为具体的评论内容。对于每一条评论，去除评论中的标点符号以及特殊字符，然后使用 jieba 分词工具进行分词。分词后的单词用空格分隔。

```
   if len(line) == 2:
        continue
```

如果是不带标签的数据，那么仅进行分词就结束

```
    if line['label'] == 'positive':
        labels.append(1)
    elif line['label'] == 'neutral':
        labels.append(0)
```

```
    elif line['label'] == 'negative':
        labels.append(2)
```

如果是带标签的数据，那么对其附加标签。其中，积极赋予 1，中性赋予 0，消极赋予 2。

```
def train_model(path):
```

该函数用于训练分类器。核心实现如下：

```
    text = read_json(path)
    words, labels = analyzes(text)
```

读入数据，进行标签分类。

```
    tf = TfidfVectorizer()
    x_train = tf.fit_transform(words).toarray()
```

将分词过的句子进行向量化，作为贝叶斯分类器的 X 向量

```
    y_train = labels
```

Y 向量为词语的情感标签

```
    mt = MultinomialNB(alpha=0.1)
    mt.fit(x_train, y_train)
    return tf, mt
```

训练分类器，并将分类器返回。

主函数如下：

```
tf, mt = train_model('train_data.json')
```

训练分类器

```
test_text = read_json('test.json')
test_words = analyzes(test_text)
```

处理待分类的数据

```
x_test = tf.transform(test_words).toarray()
predict = mt.predict(x_test)
```

使用分类器对待分类的数据进行情感分析

```
ans = []
for i in range(len(predict)):
    ans.append([i + 1, int(predict[i])])
f = open('1180300315-周牧云.csv', 'w', encoding='utf-8', newline='')
csv_writer = csv.writer(f)
csv_writer.writerows(ans)
f.close()
```

将答案写入 csv 中

```
print("finish")
```

五、实验结果分析

实验结果已提交，正确率在 69%左右。正确率最高的同学正确率能上 80%+，不知道使用了什么方法，肯定比我高明得多。

五、结论

实验的正确率大体在可接受范围内，但还有继续提高的潜力。具体的内容上，我认为可以增加更多的训练样本来训练分类器，使其分类结果更准确。在对数据进行分词时，我选择去除了非中文的符号来排除一定的干扰，但有的时候标点符号也可能附带了一些情绪因素在里面，这方面还需要进行更细致的分类。除此之外，或许我可以考虑学习下使用深度学习的工具来进行分词，例如 LSTM,准确率应该能进一步上升。