【NLP实战系列】朴素贝叶斯文本分类实战

原创 小Dream哥 有三AI 2019-10-20

收录于话题
#自然语言处理

57个

> 实战是学习一门技术最好的方式，也是深入了解一门技术唯一的方式。因此，NLP专栏计划推出一个实战专栏，让有兴趣的同学在看文章之余也可以自己动手试一试。
>
> 本篇介绍自然语言处理中一种比较简单，但是有效的文本分类手段：朴素贝叶斯模型。
>
> 作者&编辑 | 小Dream哥

1 朴素贝叶斯介绍

贝叶斯决策论是在统计概率框架下进行分类决策的基本方法。对于分类任务来说，在所有相关概率都已知的情况下，贝叶斯决策论考虑如何基于这些概率和误判损失来预测分类。

朴素贝叶斯模型在训练过程，利用数据集D，计算P(c)，P(x_i|c)。在预测时，输入样本，利用贝叶斯公式，计算n个类别的概率，最后输出概率最大的那个类别，作为预测的类别。

$$P(c_j)\prod_{i=1}^{n}P(x_i \mid c_j)$$

朴素贝叶斯模型分类的理论相关知识，在文章**【NLP】经典分类模型朴素贝叶斯解读**中有详细的介绍，感兴趣或者不清楚的朋友可以出门左转，再看一下。

假如我们有语料集D，文本可分为（c_1，c_2，...，c_n）个类别，我们根据语料集D，计算每个类别出现的概率P(c_i)，以及当文本类别为c_i时，词x_i出现的概率P(x_i|c_i)。这样一个由m个词构成的文本（x_1，x_2，...，x_m）就可以根据上述公式预测出文本是各个类别的概率。

2 NLTK

Natural Language Toolkit，NLTK是一个开源的项目，包含：Python模块，数据集和教程，用于NLP的研究和开发，是一个不错的python工具包。此次我们介绍用NLTK里的NaiveBayesClassifier模块，来进行文本分类。

3 数据集准备

因为是文本分类任务，所以数据集是按类别分好的一系列文本，如下图所示：

```
checkWeather    快转晴了吗；明天会很热吗；要下雨了吗；我想查天气；帮我查一下天
takeTaxi    叫个车；给我打个车吧；麻烦帮我打车；我想叫车；我想去平安金融大厦；
others  打球吗？；电影好看吗？不想上班啊；想去吃顿好的；有没有人去唱K呢？；外
```

这里因为只是展示，数据集比较简单，只有3个类别。在一个类别内，每条数据之间用分号隔开。

4 数据读取

数据读取的代码如下所示：

```
#1 进行数据读取
def read_data(filename):
  data = []
  with open(filename, encoding='utf-8') as f:
    for line in f:
      (label, sentences) = line.split('\t')
      sentence_list = sentences.split('；')
      data.extend([(sentence,label) for sentence in sentence_list if sentence])
# 最后返回的是一个列表，结构如下[('我要打车','get_a_taxi')，('明天天气怎么
样','get_weather')。。。]
  return data
```

数据读取过程的任务很简单，就是从语料文件中将语料读到内存中，组织成一个列表，列表中每一项组成为（data，label），如('明天天气怎么样','get_weather')。

5 特征选择及训练数据集构建

利用分词后的词性作为训练数据特征。最后喂给模型的数据是一个**类似这样的列表**：

[({“t”:“明天”,“n”：“天气”,“r”:“怎么样” }, 'get_weather') ,...]

```
#2.1 停用词处理
def delte_stop_word(sentence):
  for word in stop_word:
     if word in sentence:
        sentence.replace(word, '')
  return sentence

#2 进行特征选择，这里利用分词后的词性作为特征
def get_word_features(sentence):
  data = {}
  sentence = delte_stop_word(sentence)
  seg_list = pesg.cut(sentence)
  for word, tag in seg_list:
     data[tag] = word
  return data

#3 构建训练数据集
def get_features_sets(datafile):
  feature_sets = []
  for sentence, label in read_data(datafile):
     feature = get_word_features(sentence)
     feature_sets.append((feature, label))
  return feature_sets
```

6 训练及预测

训练及预测的过程很简单，就是调用NLTK的NaiveBayesClassifier模块，代码如下：

```
#训练模型
classifier = nltk.NaiveBayesClassifier.train(
                get_features_sets('data.txt') )

#预测某一个文本的类别
predict_label = classifier.classify(get_word_features('请问明天的天气怎么样？') )
print(predict_label)    ## get_weather

#预测某一个文本为某一个类别的概率print(classifier.prob_classify(get_word_features('请问明天的天气怎么样？')).prob(predict_label)) # 0.995154
```

我们展示一下预测的结果：

```
Loading model from cache C:\Users\27842\AppData\Local\Temp\jieba.cache
Loading model cost 0.693 seconds.
Prefix dict has been built succesfully.
checkWeather
0.9951548666616453
请输入您要预测的句子：
请问明天的天气怎么样？
文本<请问明天的天气怎么样？>预测类别为：checkWeather 概率为 0.995155
请输入您要预测的句子：
我想叫个出租车
文本<我想叫个出租车>预测类别为：takeTaxi 概率为 0.996392
请输入您要预测的句子：
今天会下雨吗？
文本<今天会下雨吗？>预测类别为：checkWeather 概率为 0.969303
请输入您要预测的句子：
```

至此，介绍了如何利用NLTK的NaiveBayesClassifier模块进行文本分类，代码在我们有三AI的github可以下载：

https://github.com/longpeng2008/yousan.ai/tree/master/natural_language_processing

找到intention文件夹，执行python3 intent_recognition.py就可以运行了。

总结

文本分类常常用于**情感分析、意图识别**等NLP相关的任务，是一个非常常见的任务，**朴素贝叶斯本质上统计语料中对应类别中相关词出现的频率，并依此来预测测试文本。**总的来说，它是一种非常便捷，效果可以接受的方法。

我们也会在**知识星球**讨论其他文本分类方法，感兴趣扫描下面的二维码了解。

读者们可以留言，或者加入我们的**NLP群**进行讨论。感兴趣的同学可以微信搜索**jen104**，备注"**加入有三AI NLP群**"。

下期预告：命名实体识别实践

知识星球推荐