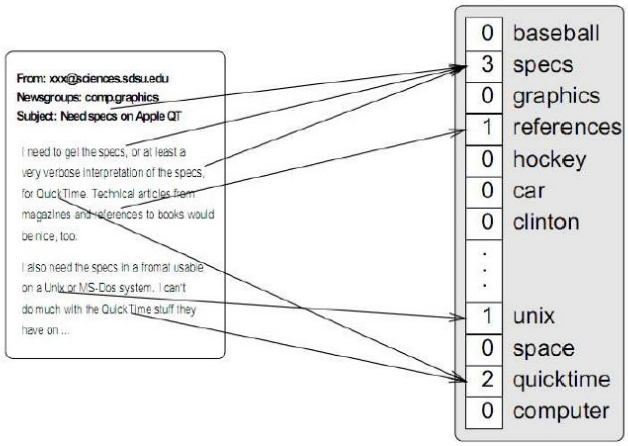
夜与周公

文本挖掘之文本表示

2013-07-25 16:56 夜与周公 阅读(9686) 评论(11) 编辑 收藏

当我们尝试使用统计机器学习方法解决文本的有关问题时，第一个需要的解决的问题是，如果在计算机中表示出一个文本样本。一种经典而且被广泛运用的文本表示方法，即向量空间模型(VSM)，俗称“词袋模型”。

我们首先看一下向量空间模型如何表示一个文本：



空间向量模型需要一个“字典”：文本的样本集中特征词集合，这个字典可以在样本集中产生，也可以从外部导入，上图中的字典是[baseball, specs, graphics,..., space, quicktime, computer]。

有了字典后便可以表示出某个文本。先定义一个与字典长度相同的向量，向量中的每个位置对应字典中的相应位置的单词，比如字典中的第一个单词baseball，对应向量中的第一个位置。然后遍历这个文本，对应文本中的出现某个单词，在向量中的对应位置，填入“某个值”。

实际上填入的“某个值”，就是当前特征词的权重(Term Weight)，目前特征词的权重主要有以下四种：

- Bool（presence）

表示某个单词是否在某个文档中出现，如果出现则记为1，否定则记为0。

$$w_{ki} = \begin{cases} 1, & if\ t_i\ exsits\ in\ d_k \\ 0, & otherwise \end{cases}$$

- Term frequency(TF)

表示某个单词在文本中出现的次数(上图中使用的权重)，一个文本中，某个特征词出现的愈多，可能其在样本中的贡献越大。

$$w_{ki} = \#(t_k, d_j)$$

- Inverse document frequency(IDF)

document frequency表示特征词在数据集中出现的文档频率。某个词文档频率越低，相应的这些文档，越容易被捕获。

About

昵称： 夜与周公
园龄： 7年7个月
粉丝： 72
关注： 3
+加关注

SEARCH

最新评论

Re:文本挖掘之特征选择(python 实现)
@lhysh怎么跑出来的，我这个出现了些问题。可以帮我解决吗... -- 20xingkong

Re:文本挖掘之特征选择(python 实现)
@祁祺 请问结果始终都是一样的这个问题是怎么解决的呢？ ... -- lhyshhsrk

Re:文本挖掘之文本表示
这篇文章写得通俗易懂，感谢。。 -- 常山之蛇

Re:文本挖掘之文本表示
谢谢,有帮助呢. -- 孤竹孙

Re:文本挖掘之特征选择(python 实现)
@ 祁祺换个数据库吧，我也是这种情况... -- 紫茉莉花开半夏

日历

< 2020年11月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

随笔档案

2014年3月(2)
2013年10月(1)
2013年8月(9)
2013年7月(3)
2013年6月(6)
2013年5月(9)
2013年4月(1)
2013年3月(5)

随笔分类

C++(13)
Python(3)
机器学习(13)
算法(14)
文本挖掘与情感分析(2)

推荐排行榜

1. 文本挖掘之文本表示(7)
2. 文本挖掘之特征选择(python 实现)(5)
3. logistic regression C++实现(2)
4. 熵、信息增益以及其他(1)
5. 寻找最大(小)的K个数(1)

阅读排行榜

1. 文本挖掘之特征选择(python 实现)(29398)
2. 文本挖掘之文本表示(9686)
3. 逻辑斯特回归模型(logistic regression)(6808)
4. logistic regression C++实现(4374)
5. 多分类问题与多类感知机算法(2790)

$$w_{ki} = \log \frac{N}{df_i}$$

- TF-IDF

TF-IDF则综合了上面两种特征权重的性质。

$$w_{ki} = \#(t_i, d_j) \log \frac{N}{df_i}$$

有关于“教育”的文档中，“高校”、“学生”等词出现的频率很高，而在“体育”类的文档中，“比赛”，“选手”出现的频率比很高。采用TF权重，这些特征词有着较高权重是合理的(Term frequency)。但是，某些词如“这些”，“是”，“的”，也有着较高的词频，但是重要度显然没有，“高校”、“学生”、“比赛”，“选手”来得重要。但“这些”，“是”，“的”这些词IDF往往比较低，很好的弥补了TF的缺陷。因此TF-IDF权重，在传统的文本分类，信息检索领域有着非常广泛的运用。

尽管TF-IDF权重有着非常广泛的应用，并不是所有的文本权重采用TF-IDF都会有较好的性能。比如，情感分类(Sentiment Classification)问题上，采用BOOL型的权重往往有较好的性能(Sentiment Classification的很多论文都采用BOOL型权重)。

现在，我们回到文章开头提高的向量空间模型。基于向量空间模型表示方法，每个特征词之间相互独立。由于这种表示简单的特点，在开始之初，推动了文本分类相关研究工作，但是随着时间的推移，传统的向量空间模型由于丢弃了词序、句法和部分语义信息，往往限制了某些领域的发展(如Sentiment Classification)，成为影响性能的瓶颈。目前的解决思路有：

- 使用N-Gram语法特征
- 将语法语义信息考虑到分类任务中
- 模型上改进...

最后，介绍一下sklearn中的文本的表示方法，并以此实现一个简单的文本分类。

我们使用的数据集是 movie_reviews语料(情感分类器任务)。数据集的组织方式是，一个文本存放在文件下，标签相同的文件放在同一个文件夹下。其数据集的结构如下：

movie_reviews\

pos\

cv000_29590.txt, cv001_18431.txt...cv999_13106.txt

neg\

cv000_29416.txt, cv001_19502.txt...cv999_14636.txt

在sklearn中，sklearn.datasets.load_files，可以很好的加载这种结构的数据集，数据加载完成后，就可以利用前面介绍的VSM，将文本样本表示出来。

sklearn专门提供了文本特征的提取模块:sklearn.feature_extraction.text，完成将一个文本样本变成一个词袋。CountVectorizer对应词频权重或是BOOL型权重(通过参数binary调节)向量空间模型， TfidfVectorizer提供了Tfidf权重下的向量空间模型。sklearn为他们提供了大量的参数（所有参数也都提供了默认参数），具有很高的灵活性和实用性。

在movie_reviews语料上，基于 sklearn 文本表示方法，并使用Multinomial Naive Bayes分类器进行情感分类的代码如下：

```
#!/usr/bin/env python
# coding=gbk

import os
import sys

import numpy as np
from sklearn.datasets import load_files
from sklearn.cross_validation import train_test_split
from sklearn.feature_extraction.text import  CountVectorizer
from sklearn.naive_bayes import MultinomialNB

def text_classifly(dataset_dir_name):
    #加载数据集，切分数据集80%训练，20%测试
    movie_reviews = load_files(dataset_dir_name)
    doc_terms_train, doc_terms_test, doc_class_train, doc_class_t

    #BOOL型特征下的向量空间模型，注意，测试样本调用的是transform接口
    count_vec = CountVectorizer(binary = True)
    doc_train_bool = count_vec.fit_transform(doc_terms_train)
    doc_test_bool = count_vec.transform(doc_terms_test)

    #调用MultinomialNB分类器
    clf = MultinomialNB().fit(doc_train_bool, doc_class_train)
    doc_class_predicted = clf.predict(doc_test_bool)

    print 'Accuracy: ', np.mean(doc_class_predicted == doc_class_


if __name__ == '__main__':
    dataset_dir_name = sys.argv[1]
    text_classifly(dataset_dir_name)
```


夜与周公
关注 - 3
粉丝 - 72
+加关注

7 0

« 上一篇： 寻找序列中满足条件的元素
» 下一篇： 亲和数问题

分类 Python，机器学习，文本挖掘与情感分析

#1楼 polymorphic
2013-07-25 17:27

ADD YOUR COMMENT

看起来好强大啊，但看不懂。
学习了，尤其是TF-IDF权重，我怎么就没想到呢?
另外请教一下，我们做OCR识别-匹配药品名称，就是把一张发票上的药品用OCR识别出来，但匹配上一直用的Levenshtein算法，不知道有没有更好的?

支持(0) 反对(0)