文本挖掘系统介绍

Introduction of Text Mining System

李宁

http://www.lining0806.com/

文本挖掘

- 词频统计
- 词性标注
- 文本分类
- 标签提取
- 实体发现和识别
- 情感趋势分析

实现目标

苹果要在印度建立4000人的研发中心，专门开发苹果地图

2016年05月19日15:23 新浪科技 微博 我有话说 收藏本文

新浪科技讯 北京时间5月19日下午消息，为了进一步吸引印度用户和开发者，苹果CEO蒂姆·库克(Tim Cook)周四宣布在印度南部的海德巴拉建设一个新的研发中心，专门开发该公司的地图产品。

苹果周三早些时候宣布，将于明年初在印度班加罗尔建设一处新的设施，专门帮助开发者采用最佳的开发模式，并改进iOS应用的设计、质量和性能。

库克目前正在进行对印度的第一次访问，而在iPhone全球销量和整个公司的营收双双下滑的背景下，苹果iPhone今年第一季度在该国实现56%的销量增长。

苹果的这个新研发中心将专注于为iPhone、iPad、MAC和Apple Watch等产品开发地图功能。苹果表示，这笔投资将加快苹果地图的开发速度，并在当地创造4000个就业岗位。

苹果并未披露这笔投资的具体规模，但有报道称可能达到2500万元。

科技？财经？娱乐？匹配关键词靠谱不？

科技：{电脑，互联网，…}

财经：{股票，黄金，白银，…}

娱乐：{游戏，音乐，电影，…}

问题1：关键词重复？

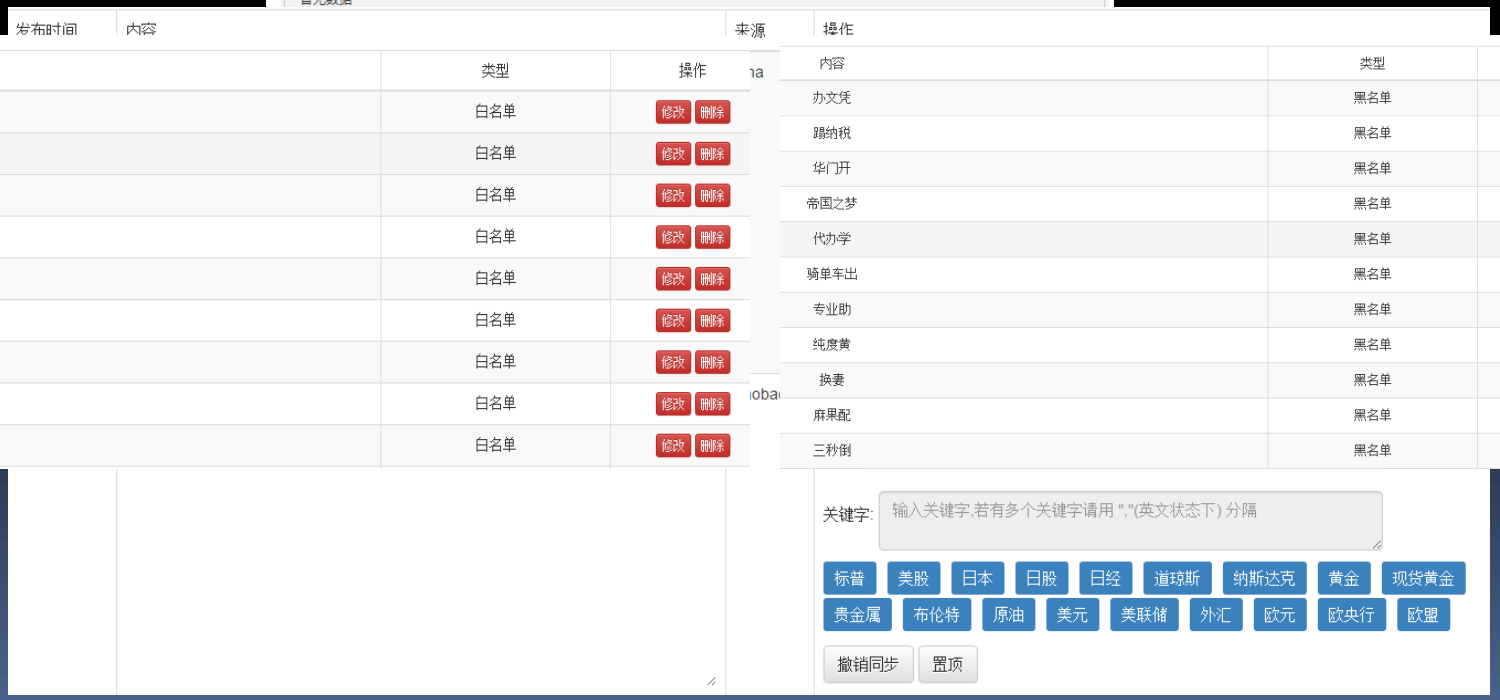
问题2：新生词语的出现？

问题3：“苹果公司开始进军游戏行业，股价进一步上涨。”

实现目标

内容	类型	操作
耶伦	白名单	修改 删除
日本	白名单	修改 删除
日经	白名单	修改 删除
道琼斯	白名单	修改 删除
原油	白名单	修改 删除
外汇	白名单	修改 删除
欧盟	白名单	修改 删除
美股	白名单	修改 删除
布伦特	白名单	修改 删除

内容	类型	操作
办文凭	黑名单	修改 删除
踢纳税	黑名单	修改 删除
华门开	黑名单	修改 删除
帝国之梦	黑名单	修改 删除
代办学	黑名单	修改 删除
骑单车出	黑名单	修改 删除
专业助	黑名单	修改 删除
纯度黄	黑名单	修改 删除
换妻	黑名单	修改 删除
麻果配	黑名单	修改 删除
三秒倒	黑名单	修改 删除



实现流程

主词典文件

李小福是创新办主任也是云计算方面的专家

李小福 / 是 / 创新 / 办 / 主任 / 也 / 是 / 云 / 计算 / 方面 / 的 / 专家 /

李小福 / 是 / 创新办 / 主任 / 也 / 是 / 云计算 / 方面 / 的 / 专家 /

分词白名单

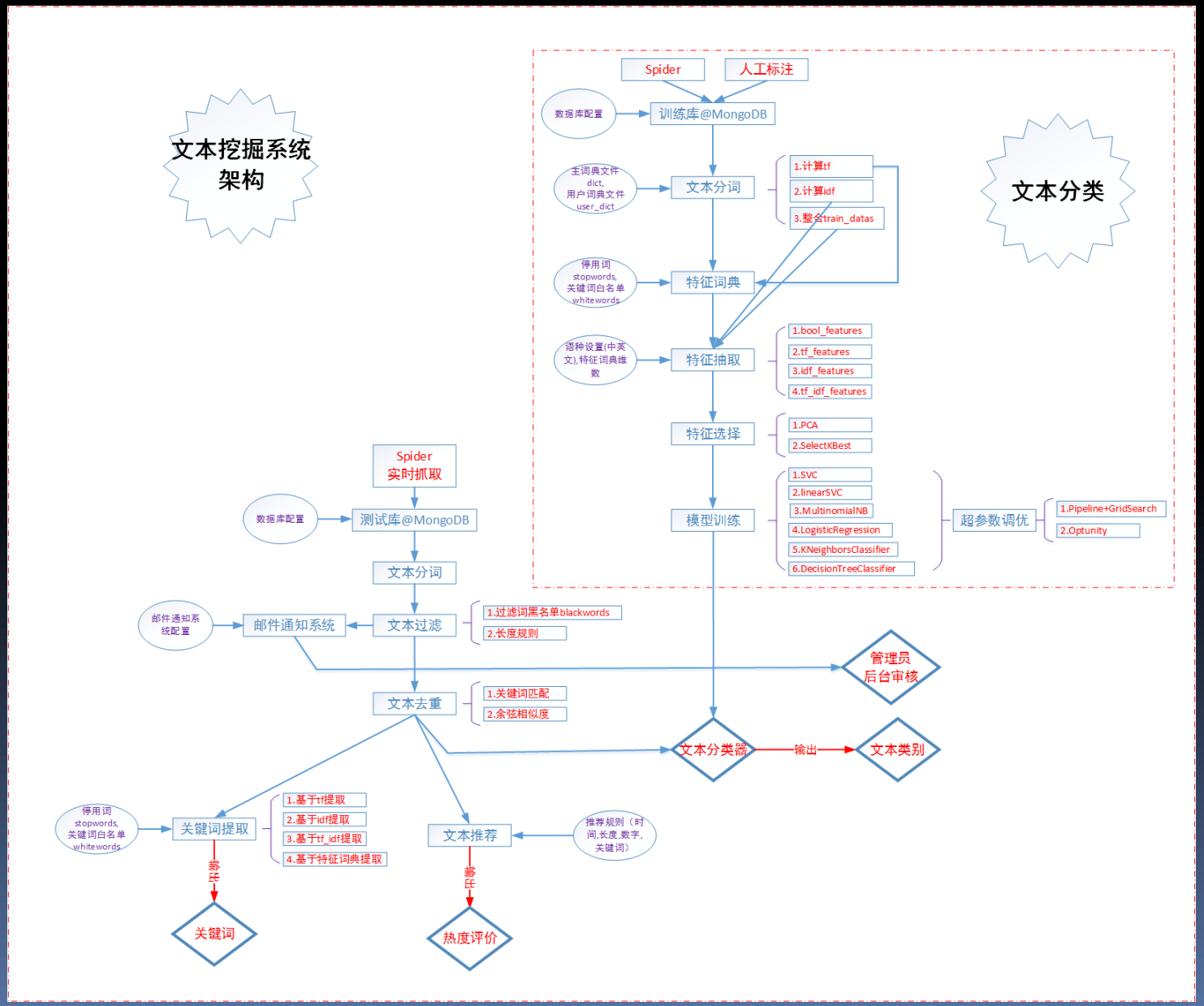
小结:

1. “自学习”功能
2. 扩展到其他应用

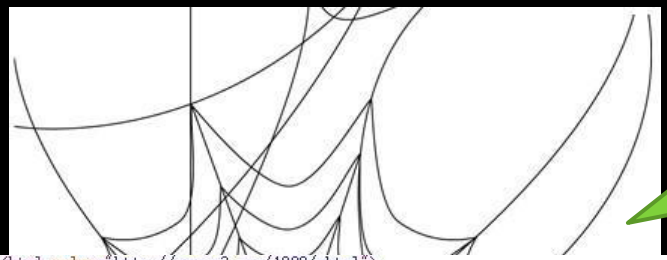
我们如何去做优化?

1. 分词白名单（一般不搞）
2. 过滤词黑名单及过滤规则的总结
3. 停用词及关键词白名单
4. 分类统计的优化（机器学习）
5. 推荐规则的总结

系统框架


文本挖掘系统架构
文本分类
Spider
人工标注
数据库配置
训练库@MongoDB
主词典文件 dict,
用户词典文件 user_dict
文本分词
1.计算tf
2.计算idf
3.整合train_datas
停用词 stopwords,
关键词白名单 whitewords
特征词典
语种设置(中英文),特征词典维数
特征抽取
1.bool_features
2.tf_features
3.idf_features
4.tf_idf_features
特征选择
1.PCA
2.SelectKBest
模型训练
1.SVC
2.linearSVC
3.MultinomialNB
4.LogisticRegression
5.KNeighborsClassifier
6.DecisionTreeClassifier
超参数调优
1.Pipeline+GridSearch
2.Optunity
Spider
实时抓取
数据库配置
测试库@MongoDB
文本分词
邮件通知系统配置
邮件通知系统
文本过滤
1.过滤词黑名单blackwords
2.长度规则
管理员后台审核
文本去重
1.关键词匹配
2.余弦相似度
文本分类器
输出
文本类别
停用词 stopwords,
关键词白名单 whitewords
关键词提取
1.基于tf提取
2.基于idf提取
3.基于tf_idf提取
4.基于特征词典提取
蒙版
关键词
文本推荐
推荐规则（时间,长度,数字,关键词）
蒙版
热度评价

数据抓取——Spider



指的是通过程序实现访问某个URL地址，然后获得其所返回的内容（HTML源码，Json格式的字符串等）。然后通过解析规则，分析出我们需要的数据并取出来。

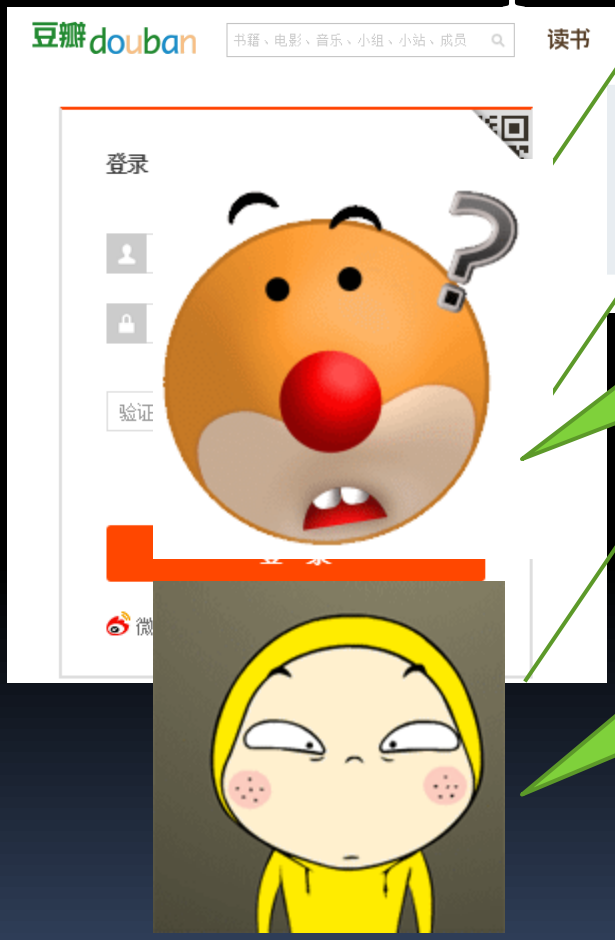
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <meta http-equiv="Content-Language" content="zh-CN" />
    <title>新闻排行榜_网易新闻</title>
    <base target="_blank" />
    <meta name="keywords" content="" />
    <meta name="description" content="" />
    <meta name="author" content="网易" />
    <meta name="Copyright" content="网易版权所有" />
    <script>if(!/auto|house|home|bbs|blog/.test(location.host)&&!(document.documentElement&&document.documentElement.getAttribute("phone"))&&!/_pc=1/.test(location.
mobile|ipod|blackberry|bb\d+|phone/i.test(navigator.userAgent))document.write('<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"
<div style="position:absolute;top:50%;left:0;width:100%;height:40px;margin-top:-40px;text-align:center;background:url(http://img1.cache.netease.com/utf8/endpage/ima
载中 ...</div></div><script src="http://img1.cache.netease.com/f2e/system/touchall/collect/foot~3Cwae6PoOSne.js"'+' defer><'+'/script><plaintext style="display:none
<script >var _ntes_const={stime : new Date()};</script>
    <link href="http://img1.cache.netease.com/cnews/css07/style.css" rel="stylesheet" type="text/css" />
    <link href="http://img1.cache.netease.com/cnews/img09/channel_nav.css" rel="stylesheet" type="text/css"/>
    <link href="http://img1.cache.netease.com/cnews/img/subscibe0304/rank.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        .gg735 {width:735px; overflow:hidden; float:left;}
        .gg210 {width:210px; overflow:hidden; float:right;}
        .channel h2 { width: auto; }
        .subNav a.photoset-icon { background: url(http://img1.cache.netease.com/cnews/img10/bbs0114/photoset.gif) no-repeat; width:128px; }
    </style>
    <script type="text/javascript" language="javascript" src="http://img1.cache.netease.com/cnews/js/ntes_jslib_1.x.js" charset="gb2312"></script>

<script language="javascript" type="text/javascript" src="http://202.102.100.100/35ff706fd57d11c141cdefcd58d6562b.js" charset="gb2312"></script><script type="text/j
vvvvvvvvvv('.btm-ad,.gg300,.content-ad,.gg,[class$="bottom_ad"],[class~="ad2"],.ggarea,.top-gg,#addiv,[class~="ad_"],.top-gg-area,[class~="gg2"],[class~="ad ad-"],
.youdao,iframe[src^="http://g.163.com/"][src*="&affiliate="],iframe[src^="http://img1.126.net/"],iframe[src^="http://x.jd.com/"],img[src^="http://img1.126.net/"],if
<body>
<script>if(!/auto|house|home|bbs|blog/.test(location.host)&&!(document.documentElement&&document.documentElement.getAttribute("phone"))&&!/_pc=1/.test(location.sear
mobile|ipod|blackberry|bb\d+|phone/i.test(navigator.userAgent))document.write('<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"
<div style="position:absolute;top:50%;left:0;width:100%;height:40px;margin-top:-40px;text-align:center;background:url(http://img1.cache.netease.com/utf8/endpage/ima
载中 ...</div></div><script src="http://img1.cache.netease.com/f2e/system/touchall/collect/foot~3Cwae6PoOSne.js"'+' defer><'+'/script><plaintext style="display:none
<link type="text/css" rel="stylesheet" media="screen" href="http://img1.cache.netease.com/common/css/common_nav_v1.0.8.css" />
<style>
```

https://github.com/lining0806/PythonSpiderNotes

数据抓取——Spider

取

存



使用Cookie登录

AJAX?
反盗链?
断线重连?
……

终极武器:
Selenium！！

解析规则：正则表达式，xpath，csspath,……

数据抓取——Spider

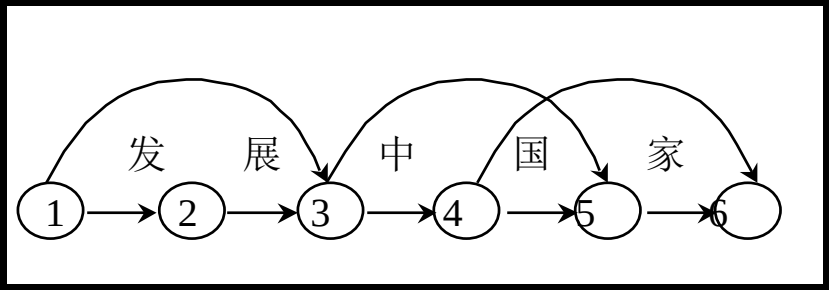
微信公众号自动登录，抓取图文统计数据
——技术难点：
1.Selenium获取动态token
2.Api的时效性

指定关键词搜索微信文章并存入数据库

https://github.com/lining0806/PythonSpiderNotes

中文分词

分词就是将连续的字串或字符序列按照一定的规范重新组合成词序列的过程。



主要的分词方法

- ◆ 简单的模式匹配:
- 正向最大匹配
- 逆向最大匹配
- 双向匹配法
- ◆ 基于规则的方法:
- 最少分词算法
- ◆ 基于统计的方法:
- 统计语言模型分词
- 串频统计和词形匹配相结合
- 无词典分词

https://github.com/fxsjy/jieba

特征抽取

TF-IDF的主要思想

如果某个词或短语在一篇文章中出现的频率高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

TF (Term Frequency)：词频

IDF (Inverse Document Frequency)：逆文档频率

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$\mathrm{idf}_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

http://www.ruanyifeng.com/blog/2013/03/tf-idf.html

特征选择

特征选择方法

◆ 去除无用特征:

- 不必要的特征对训练无用。

◆ 去除相关分量:

- 相关的多个分量可以变换成较少的不相关分量。

PCA（Principal Component Analysis）

- 主成分分析是设法将原来众多具有一定相关性（比如P个指标），重新组合成一组新的互相无关的综合指标来代替原来的指标。

假定有n个样本，每个样本共有p个变量，

构成一个$n \times p$阶的数据矩阵

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

降维处理

$$\begin{cases} z_1 = l_{11}x_1 + l_{12}x_2 + \cdots + l_{1p}x_p \\ z_2 = l_{21}x_1 + l_{22}x_2 + \cdots + l_{2p}x_p \\ \vdots \\ z_m = l_{m1}x_1 + l_{m2}x_2 + \cdots + l_{mp}x_p \end{cases}$$

http://www.cnblogs.com/LeftNotEasy/archive/2011/01/08/lda-and-pca-machine-learning.html

分类器设计

朴素贝叶斯（Naive Bayes）

Naive Bayes是一个生成模型，在计算P(y|x)之前，先要从训练数据中计算P(x|y)和P(y)的概率，从而利用贝叶斯公式计算P(y|x)。

Naive Bayes满足P(y=1|x) = P(y=1)*P(x|y=1)/p(x)

采用使得后验概率P(y|x)最大的输出，作为最佳的输出y。

$$
\begin{aligned}
P(X|Y) &= P(X_1, X_2|Y) \\
&= P(X_1|X_2, Y)P(X_2|Y) \\
&= P(X_1|Y)P(X_2|Y)
\end{aligned}
$$

$$P(X_1 \dots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

$$P(Y = y_k|X_1 \dots X_n) = \frac{P(Y = y_k)P(X_1 \dots X_n|Y = y_k)}{\sum_j P(Y = y_j)P(X_1 \dots X_n|Y = y_j)}$$

$$P(Y = y_k|X_1 \dots X_n) = \frac{P(Y = y_k)\prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i|Y = y_j)}$$

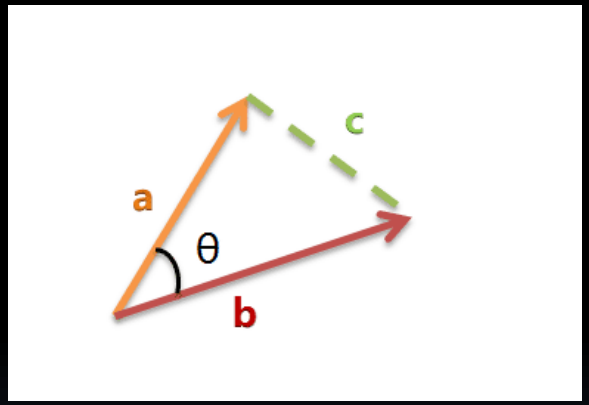
$$Y \leftarrow \arg\max_{y_k} \frac{P(Y = y_k)\prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i|Y = y_j)}$$

$$Y \leftarrow \arg\max_{y_k} P(Y = y_k)\prod_i P(X_i|Y = y_k)$$

http://www.ruanyifeng.com/blog/2013/12/naive_bayes_classifier.html

文章相似度

余弦相似度



$$\cos\theta = \frac{a^2 + b^2 - c^2}{2ab}$$

http://www.ruanyifeng.com/blog/2013/03/cosine_similarity.html

更多资源

项目链接：https://github.com/lining0806/TextMining

```
    ## ------------------------------------------------------------------------------------
    ## 生成分类器模型
    feature_selection_flag = False
    my_selector = None
    if test_speedup and os.path.exists(fea_dict_file) and os.path.exists(best_clf_file):
        words_feature = []
        with open(fea_dict_file, 'r') as fp:
            for line in fp.readlines():
                word_feature = line.strip().decode("utf-8")
                words_feature.append(word_feature)
        if feature_selection_flag:
            with open(best_clf_file, "rb") as fp_pickle:
                my_selector, best_clf = pickle.load(fp_pickle)
        else:
            with open(best_clf_file, "rb") as fp_pickle:
                best_clf = pickle.load(fp_pickle)
    else:
        ## ------------------------------------------------------------------------------------
        words_feature = MakeFeatureWordsDict(all_words_tf_dict, stopwords_set, writewords_set, lag, fea_dict_size)
        train_features = []
        train_class = []
        for train_data in train_datas:
            TextFeatureClass = TextFeature(words_feature, train_data[0])
            train_features.append(TextFeatureClass.TextBool()) #### 可以调整特征抽取，训练集与测试集保持一致
            train_class.append(int(train_data[1])) # str转为int
        train_features = np.array(train_features)
        train_class = np.array(train_class)
        if feature_selection_flag:
            FeatureSelectorClass = FeatureSelector(train_features, train_class)
            my_selector, train_features = FeatureSelectorClass.PCA_Selector() #### 可以调整特征选择
        start_time_train = datetime.datetime.now()
        ClassifierTrainClass = ClassifierTrain(train_features, train_class)
        best_clf = ClassifierTrainClass.LR() #### 可以调整分类器训练
        end_time_train = datetime.datetime.now()
        print "best_clf training last time:", end_time_train-start_time_train
        if not os.path.exists(Classifier_Dir):
            os.makedirs(Classifier_Dir)
        with open(fea_dict_file, 'w') as fp:
            for word_feature in words_feature:
```

谢谢！