
EXP1 文本模块大作业

汪楚文 2018202114 05/28/2020

Copyright © 2020- by Wangchuwen. All rights reserved

实验概述

[实验环境]

Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

Python 3.7

[实验数据]

1. 带标签数据（用于分类器模型训练和测试、以及检索）

标签：1表示垃圾短信 / 0表示正常短信

文本域：短信源文本（已经进行了一些必要处理）

2. 不带标签数据

[实验要求]

1. 对上述数据进行分词；
2. 基于带标签数据训练一个垃圾短信分类器，测试其分类精度；
3. 基于带标签和未带标签短信数据构造一个搜索引擎，返回结果的每一项包括
(1) 短信内容，(2) 标明返回的短信是否是垃圾短信；
4. 撰写实验报告和结果分析。

实验内容

一.分词 wordChopping.py

[代码说明]

程序编号	1	文件名	wordChopping.py	说明	用于分词
<pre>import jieba #unmark文件分词 #打开raw文件 file1 = open('./data/unmarked.txt', 'r', encoding='UTF-8') #读取 content = file1.read() #jieba分词 cut_content = jieba.cut(content) #打开result文件 file1_r = open('./result/unmarked.txt', 'w', encoding='UTF-8') #写入 file1_r.write(' '.join(cut_content)) #关闭两文件 file1.close() file1_r.close() #mark文件分词 #打开raw文件 file2 = open('./data/marked.txt', 'r', encoding='UTF-8') #读取 content2 = file2.read() #jieba分词 cut_content2 = jieba.cut(content2) #打开result文件 file2_r = open('./result/marked.txt', 'w', encoding='UTF-8') #写入 file2_r.write(' '.join(cut_content2)) #关闭两文件 file2.close() file2_r.close()</pre>					

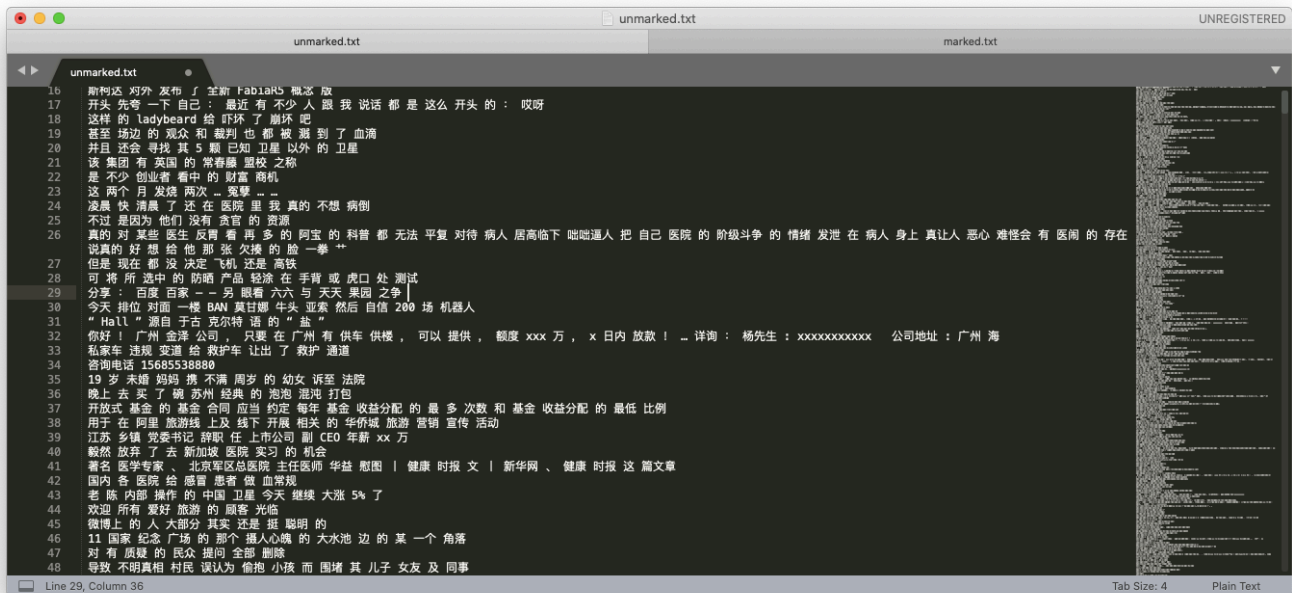
流程为：打开raw文件->读取->打开result文件->写入->关闭两文件

核心功能通过jieba.cut完成

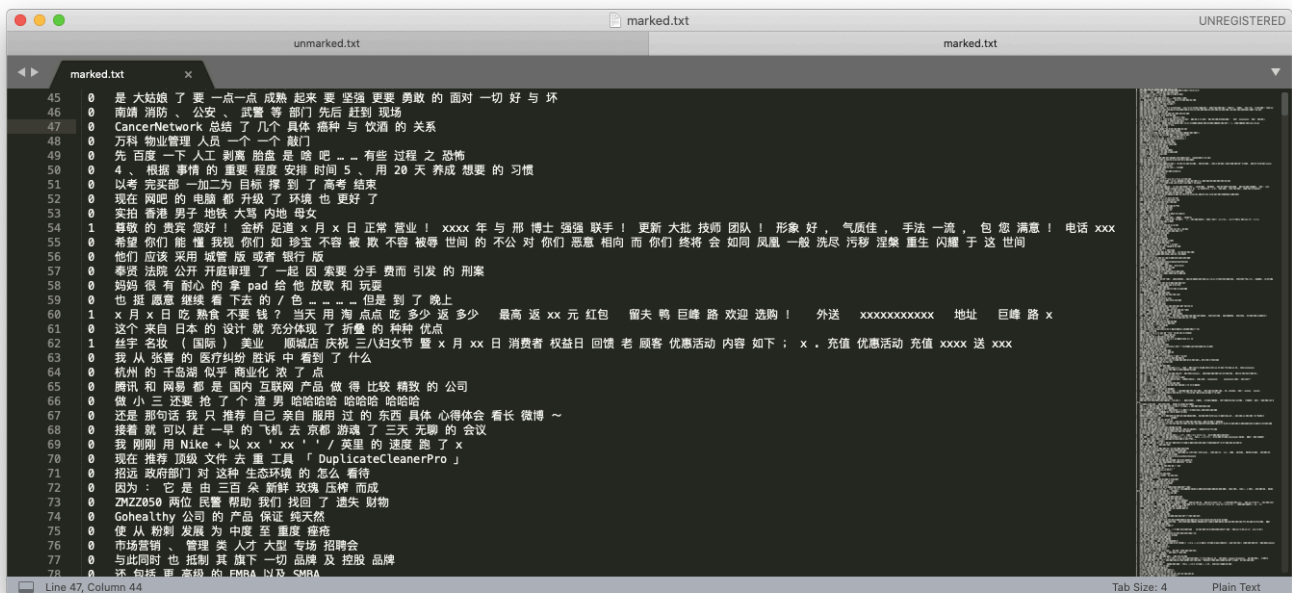
[各步骤结果分析]

对unmarked.txt和marked.txt进行jieba分词，并把jieba分词的结果储存在/result下的unmarked.txt和marked.txt中。经过检验/result下的unmarked.txt和marked.txt均成功jieba分词。

[结果示例]



```
16 斯柯达 对外 发布 了 全新 FabiaRS 概念 版
17 开头 先夸 一下 自己： 最近 有 不少 人 跟 我 说话 都 是 这么 开头 的： 哎呀
18 这样 的 ladybeard 给 吓坏 了 崩坏 吧
19 甚至 场边 的 观众 和 裁判 也 都 被 溅 到 了 血滴
20 并且 还会 寻找 其 5 颗 已知 卫星 以外 的 卫星
21 该 集团 有 英国 的 常春藤 盟校 之称
22 是 不少 创业者 看中的 财富 商机
23 这 两个 月 发险 两次 - 冤孽 - -
24 凌晨 快 清晨 了 还 在 医院 里 我 真的 不想 病倒
25 不过 是因为 他们 没有 贪官 的 资源
26 真的 对 某些 医生 反胃 看 再 多 的 阿宝 的 科普 都 无法 平复 对待 病人 居高临下 咄咄逼人 把 自己 医院 的 阶级斗争 的 情绪 发泄 在 病人 身上 真让人 恶心 难怪 会有 医闹 的 存在
27 说真的 好 想 给 他 那 张 欠揍 的 脸 一拳 - -
28 但是 现在 都 没 决定 飞机 还是 高铁
29 可 将 所 选中 的 防晒 产品 轻涂 在 手背 或 虎口 处 测试
30 分享： 百度 百家 - - 另 眼看 六六 与 天天 果园 之争 |
31 今天 排位 对 面 一 楼 DM 真甘娜 牛头 亚索 然后 自信 200 场 机器人
32 " Hall " 源自 于 古 凯尔特 语 的 " 盖 "
33 你好！ 广州 金泽 公司， 只要 在 广州 有 供车 供楼， 可以 提供， 额度 xxx 万， x 日 内 放款！ - 详询： 杨先生： xxxxxxxxxx 公司地址： 广州 海
34 私家车 违规 变道 给 救护车 让出 了 救护 通道
35 咨询电话 15685538880
36 19 岁 未婚 妈妈 携 不满 周岁 的 幼女 诉至 法院
37 晚上 去 买 了 碗 苏州 经典 的 泡泡 馄饨 打包
38 开放式 基金 的 基金 合同 应当 约定 每年 基金 收益分配 的 最 多 次数 和 基金 收益分配 的 最低 比例
39 用于 在 阿里 旅游线 上及 线下 开展 相关 的 华侨城 旅游 营销 宣传 活动
40 江苏 乡镇 党委书记 辞职 任 上市公司 副 CEO 年薪 xx 万
41 毅然 放弃 了 去 新加坡 医院 实习 的 机会
42 著名 医学专家， 北京 军区 总医院 主任医师 华益 想图 | 健康 时报 文 | 新华网、 健康 时报 这 篇文章
43 国内 各 医院 给 感冒 患者 做 血常规 提
44 老 陈 内部 操作 的 中国 卫星 今天 继续 大涨 5% 了
45 欢迎 所有 爱好 旅游 的 顾客 光临
46 微博上 的 人 大部分 其实 还是 挺 聪明 的
47 11 国家 纪念 广场 的 那个 摄人心魄 的 大水池 边 的 某 一个 角落
48 对 有 质疑 的 民众 提问 全部 删除
49 导致 不明真相 村民 误认为 偷抱 小孩 而 围堵 其 儿子 子女 及 同事
```



```
45 0 是大姑娘 了 要 一点 一点 成熟 起来 要 坚强 更 要 勇敢 的 面对 一切 好 与 坏
46 0 南靖 消防、 公安、 武警 等 部门 先后 赶到 现场
47 0 CancerNetwork 总结 了 几个 具体 癌种 与 饮酒 的 关系
48 0 万科 物业管理 人员 一个 一个 敲门
49 0 先 百度 一下 人工 剥离 胎盘 是 啥 吧 - - 有些 过程 之 恐怖
50 0 4、 根据 事情 的 重要 程度 安排 时间 5、 用 20 天 养成 想要 的 习惯
51 0 以考 完 买部 一加二 为 目标 撑 到了 高考 结束
52 0 现在 网吧 的 电脑 都 升级 了 环境 也 更好 了
53 0 实拍 香港 男子 地铁 大骂 内地 母女
54 1 尊敬的 贵宾 您好！ 金侨 足道 x 月 x 日 正常 营业！ xxxx 年 与 那 博士 强强 联手！ 更新 大批 技师 团队！ 形象 好， 气质佳， 手法 一流， 包您 满意！ 电话 xxx
55 0 希望 你们 能 懂 我 视 你们 如 珍宝 不 容 被 欺 不 容 被 辱 世间 的 不公 对 你们 恶语 相向 而 你们 终 将会 如同 凤凰 一般 洗尽 污秽 涅槃 重生 闪耀 于 这 世间
56 0 他们 应该 采用 城管 版 或者 银行 版
57 0 肇庆 法院 公开 开庭 审理 一起 因 索要 分手 费 而 引发 的 刑案
58 0 妈妈 很有 耐心 的 拿 pad 给 他 放歌 和 玩耍
59 0 也 挺 愿意 继续 看 下去 的 / 色 - - - 但是 到了 晚上
60 1 x 月 x 日 吃 熟食 不要 钱？ 当天 用 海 点 吃 多少 返 多少 最高 返 xx 元 红包 留夫 鸭 巨峰 路 欢迎 选购！ 外送 xxxxxxxxxx 地址 巨峰 路 x
61 0 这个 来自 日本 的 设计 就 充分 体现 了 折叠 的 种种 优点
62 1 丝宇 名牧 ( 国际 ) 美业 顺城 店 庆祝 三八 妇女节 暨 x 月 xx 日 消费者 权益 日 回馈 老 顾客 优惠活动 内容 如下： x . 充值 优惠活动 充值 xxxx 送 xxx
63 0 我 从 张善 的 医疗 纠纷 胜诉 中 看到 了 什么
64 0 杭州 的 千岛湖 似乎 商业 化 浓 了 点
65 0 腾讯 和 网易 都 是 国内 互联网 产品 做 得 比较 精致 的 公司
66 0 做 小 三 还 要 抢 了 个 道 男 哈哈哈哈哈 哈哈哈哈哈
67 0 还是 那句 话 我 只 推荐 自己 亲自 服用 过 的 东西 具体 心得 体会 看 长 微博 ~
68 0 接着 就 可以 赶 一早 的 飞机 去 京都 游玩 了 三天 无聊 的 会议
69 0 我 刚刚 用 Nike + 以 xx * xx + / 英里 的 速度 跑了 x
70 0 现在 推荐 顶级 文件 去 重 工具 「 DuplicateCleanerPro 」
71 0 招远 政府 部门 对 这种 生态环境 的 怎么 看待
72 0 因为： 它 是 由 三 白 朵 新鲜 玫瑰 压榨 而成
73 0 ZHZZ050 两位 民警 帮助 我们 找回 了 遗失 财物
74 0 Gohealthy 公司 的 产品 保证 纯天然
75 0 使 从 粉刺 发展 为 中度 至 重度 痤疮
76 0 市场营销、 管理 类 人才 大型 专场 招聘会
77 0 与 此 同时 也 抵制 其 旗下 一切 品牌 及 控股 品牌
78 0 这 电话 更 高级 的 PHRA LTM SHRA
```

[其他重要内容]

由于marked文件中标签是与内容隔开的，故不需要把内容单独jieba分词。

二.垃圾分类器 trashClassifier.py

[代码说明]

程序编号	2	文件名	trashClassifier.py	说明	用于垃圾分类
<pre>import pandas as pd import jieba from sklearn import metrics from sklearn.model_selection import train_test_split from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer from sklearn.naive_bayes import MultinomialNB from sklearn.metrics import accuracy_score #读取标记数据 data = pd.read_csv(r"./data/marked.txt", sep='\t', names=['label', 'text']) #jieba分词 data['cut_message'] = data['text'].apply(lambda x: ' '.join(jieba.cut(x))) #value数据集 x = data['cut_message'].values y = data['label'].values #分割训练集和数据集 train_x, test_x, train_y, test_y = train_test_split(x, y, test_size=0.1) # test_size:train_size=1:9 #向量化, 生成word_vector vectorizer = CountVectorizer() x_train_termcounts = vectorizer.fit_transform(train_x) tfidf_transformer = TfidfTransformer() x_train_tfidf = tfidf_transformer.fit_transform(x_train_termcounts) #开始训练 classifier = MultinomialNB().fit(x_train_tfidf, train_y) #向量化, 生成word_vector x_input_termcounts = vectorizer.transform(test_x) x_input_tfidf = tfidf_transformer.transform(x_input_termcounts) #预测 predicted_categories = classifier.predict(x_input_tfidf) #测试其分类精度,classification_report与confusion_matrix print(accuracy_score(test_y, predicted_categories)) print(metrics.classification_report(test_y, predicted_categories)) print(metrics.confusion_matrix(test_y, predicted_categories)) ''' #把unmark数据变为mark数据, 便于搜索引擎输出标签 data2 = pd.read_csv(r"./data/unmarked.txt", sep='\n', names=['text']) data2['cut_message'] = data2['text'].apply(lambda x: ' '.join(jieba.cut(x))) x_input_termcounts2= vectorizer.transform(data2['cut_message']) x_input_tfidf2 = tfidf_transformer.transform(x_input_termcounts2) predicted_categories2 = classifier.predict(x_input_tfidf2) k=open('./result/unmarked_i.txt', 'w', encoding='UTF-8') #print(predicted_categories2) f = open("./data/unmarked.txt", "r") for line in f: list(predicted_categories2).append(line) for n in list(predicted_categories2): k.write(n) f.close() k.close() '''</pre>					

核心功能通过sklearn.naive_bayes实现
重要数据结构:vectorizer.transform()

[流程及各步骤结果分析]

各步骤均正常运行。

将数据集按9:1划分为训练集和测试集：划分成功；

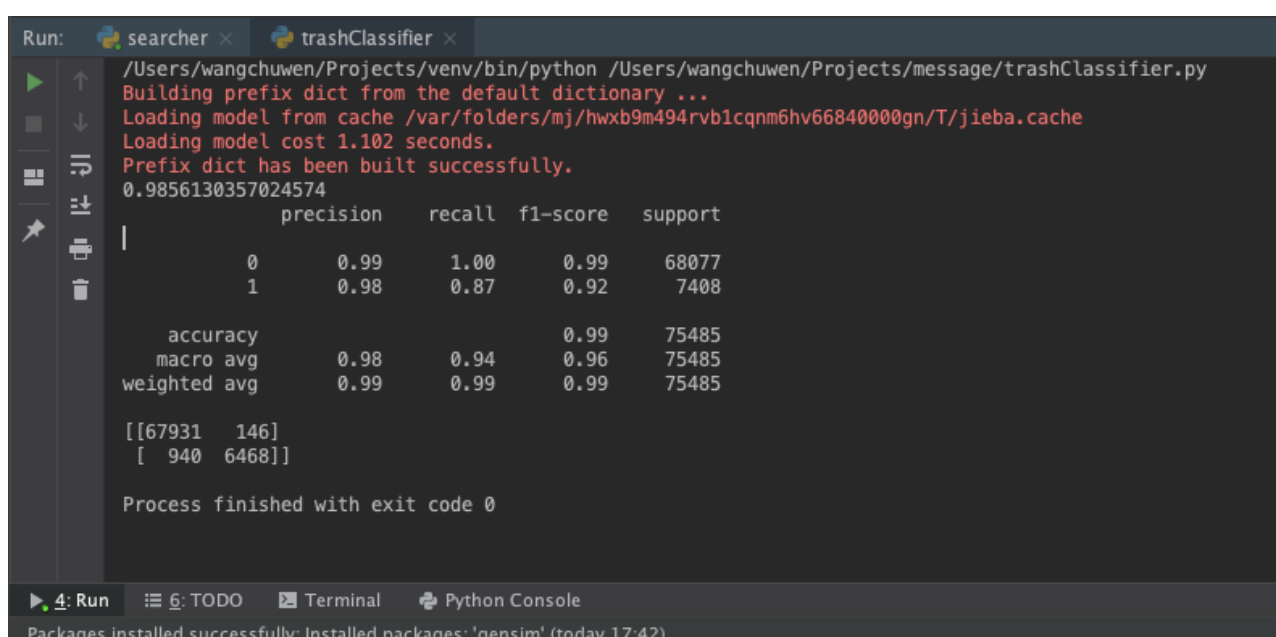
转化为word_vector: 转化成功；

训练朴素贝叶斯模型：训练完毕；

预测测试集：预测正常进行；

输出分类效果：正常进行；

[结果示例]



```
Run: searcher x trashClassifier x
/Users/wangchuwen/Projects/venv/bin/python /Users/wangchuwen/Projects/message/trashClassifier.py
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/mj/hwxb9m494rvb1cqnm6hv66840000gn/T/jieba.cache
Loading model cost 1.102 seconds.
Prefix dict has been built successfully.
0.9856130357024574
      precision    recall  f1-score   support
0         0.99      1.00      0.99      68077
1         0.98      0.87      0.92       7408

 accuracy          0.99      75485
 macro avg          0.98      75485
weighted avg          0.99      75485

[[67931  146]
 [  940 6468]]

Process finished with exit code 0

4: Run 6: TODO Terminal Python Console
Packages installed successfully: Installed packages: 'gensim' (today 17:42)
```

Precision: 0.99

precision和recall均较高，其中非垃圾信息的recall达到100%

confusion_matrix:

67931 146

940 6468

其他数据见上图；

[其他重要内容]

代码最后的注释部分对unmark中的数据进行了预测，把结果保存在了

predicted_categories2中，r然后写入了unmark_i.txt中，此时无标签信息已经变为了有标签信息，便于在搜索引擎模块输出其是否为垃圾信息。

用readcsv来readtxt文件，经测试，若将seq设置为'\t'，则在本程序中可正确readtxt。

三.搜索引擎 searcher.py

[代码说明]

程序编号	3	文件名	searcher.py	说明	用于分词
<pre>from os.path import exists import pickle from gensim.corpora import Dictionary from gensim.models import TfidfModel from gensim.similarities import SparseMatrixSimilarity from numpy import argsort #import wordChopping #unmarked_i.txt文件在trashClassifier.py中可选择生成,此时已带有label with open('./result/unmarked_i.txt', encoding='utf-8') as f1: texts1=f1.read().split('\n') #读取marked文件 with open('./data/marked.txt', encoding='utf-8') as f2: texts=f2.read().split('\n') #合并文件集合 texts=texts+texts1 # TF-IDF模型实现 PATH = 'model.pickle'#保存到pickle中, 故一旦生成pickle文件, 后续查询花费时间极小 if exists(PATH): with open(PATH, 'rb') as f: dictionary, tfidf = pickle.load(f)#若pickle文件存在则load else: corpora = [list(t) for t in texts] #建立词典 dictionary = Dictionary(corpora) tfidf = TfidfModel(dictionary.doc2bow(c) for c in corpora) #写入词典和pickle with open(PATH, 'wb') as f: pickle.dump((dictionary, tfidf), f)#dump # 搜索并rank, 输出一定数量的top num_features = len(dictionary.token2id) while True:#实现循环查询 kw = input('输入查询的内容 (输入0/1即可查询普通信息/垃圾信息) : ').strip() kw_vec = dictionary.doc2bow(list(kw))#vector_kw texts_kw = [t for t in texts if kw in t]#遍历查询 corpora_kw = [dictionary.doc2bow(list(t)) for t in texts_kw] index = SparseMatrixSimilarity(tfidf[corpora_kw], num_features) sim = index[tfidf[kw_vec]] # TF-IDF搜索 ids = argsort(-sim)[:100] #返回前100 for i in ids: print(texts_kw[i])</pre>					

重要数据结构: (dictionary, tfidf)

核心功能实现: `sim = index[tfidf[kw_vec]]`

[流程及各步骤结果分析]

打开unmarked_i.txt和marked.txt: 成功

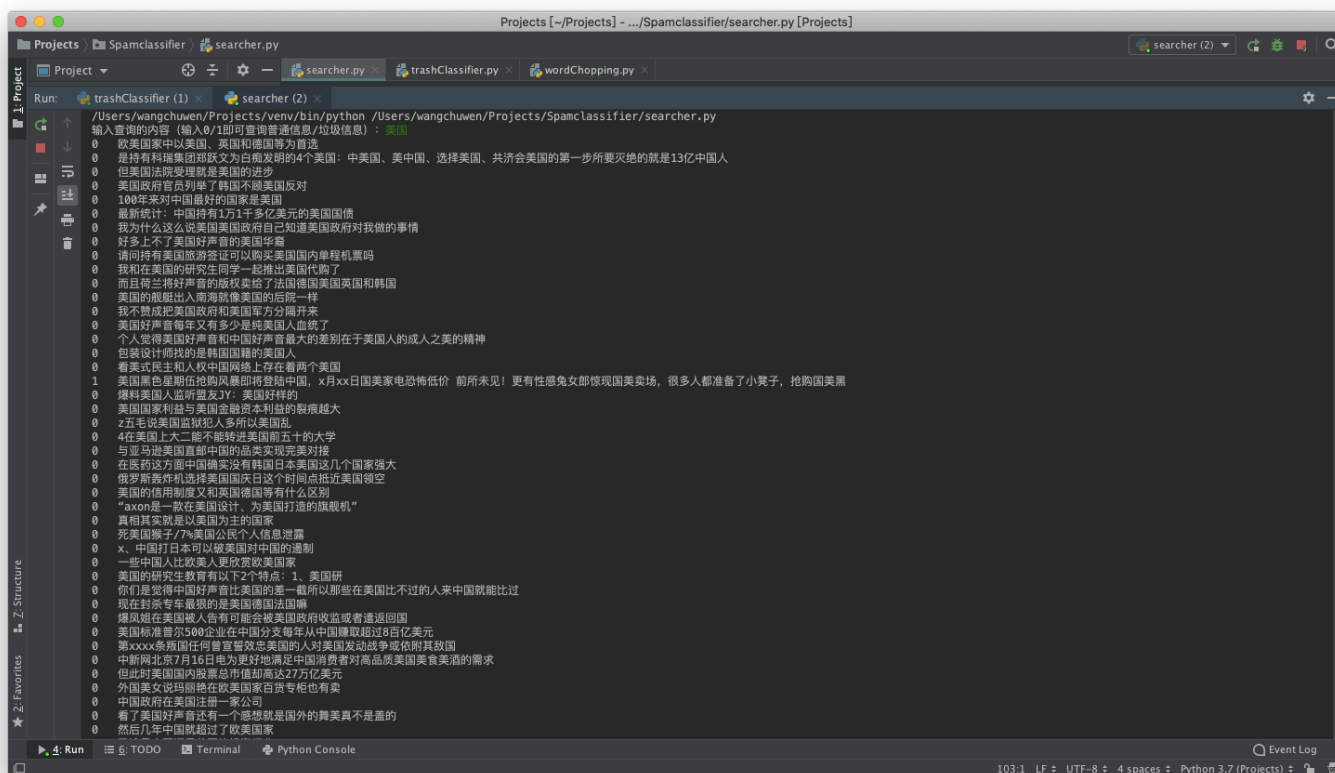
合并文件集合: 成功

读写model.pickle: 成功

查询dictionary: 成功

返回top100: 成功

[结果示例]



程序中默认返回top100, 具体可在程序中修改参数

如图: 搜索美国

[其他重要内容]

(1)在查询无标签的信息之前, 已将无标签信息unmark.txt在垃圾分类器中预测生成了有标签信息unmark_i.txt。故在此搜索引擎中, 并不是边搜索, 边预测是否为垃圾信息。

(2)若要新加入数据, 请先删除model.pickle文件, 这样model.pickle在重新生成时又会构造新的dictionary。

(3)由于pickle文件的存在, 已经建立好了倒排索引, 故搜索时间较短

目录结构

```
Spamclassifier/  
|-- data/  
|   |-- unmarked.txt  
|   |-- marked.txt  
|  
|-- result/  
|   |-- unmarked.txt  
|   |-- marked.txt  
|   |-- unmarked_i.txt  
|  
|-- model.pickle  
|  
|-- trashClassifier.py  
|-- wordChopping.py  
|-- searcher.py  
|  
|-- README.txt  
|-- 实验报告.pdf
```

[超链接： 点击此处可在github中查看该项目](#)