



# Viiision

# APP评论数据分析系统

## 技术路线及实施方案



ComVision 团队

## 目录

1. 功能模块 .....	3
1.1. 功能模块架构 .....	3
1.2. 功能模块设计 .....	3
1.2.1. 用户端 .....	3
1.2.2. 管理员端 .....	4
2. 爬虫技术 .....	5
2.1. 爬虫介绍 .....	5
2.1.1. 使用语言 .....	5
2.1.2. 数据来源分析 .....	5
2.1.3. 页面解析工具 .....	5
2.2. 爬虫框架 .....	5
2.3. 爬虫结构及部署 .....	5
2.4. 爬取数据展示 .....	6
3. 算法 .....	7
3.1. 分词算法 .....	7
3.1.1. 算法介绍 .....	7
3.1.2. 分词过程 .....	8
3.2. 关键词提取算法 .....	9
3.2.1. TF-IDF 算法介绍 .....	9
3.2.2. TF-IDF 算法使用 .....	9
3.2.3. TF-IDF 优缺点分析 .....	10
3.2.4. TF-IDF 算法改进措施 .....	10
3.3. 机器学习算法 .....	11
3.3.1. 基于朴素贝叶斯的评论分类算法 .....	11
3.3.2. 评论分类流程 .....	11
4. 垃圾评论过滤技术 .....	12
4.1.1. 垃圾评论分类算法 .....	12
4.1.2. 垃圾评论训练集的设计及更新 .....	13
5. 关键词提取技术 .....	14
5.1.1. 关键词分类 .....	14
5.1.2. 关键词提取 .....	15
5.1.3. 个性化分词添加 .....	16
6. 评论情感分析技术 .....	16
6.1. 机器学习算法 .....	16
6.1.1. 情感分析算法 .....	16
6.1.2. 评论不同极性训练集的设计及更新 .....	17
6.2. 模式匹配算法 .....	17
6.3. 混合模式 .....	17
7. 软件技术 .....	18
7.1. Laravel 框架 .....	18
7.1.1. 模型—视图—控制器 .....	18
7.1.2. 响应流程 .....	19
7.2. Nginx 服务器 .....	19
7.3. 基于 Ajax 技术的 Web 服务架构 .....	19

8. 前端交互技术 .....	20
8.1. 框架选择 .....	20
8.2. 产品颜色设计 .....	21
8.3. 响应式布局 .....	21
9. 数据可视化技术 .....	22
9.1. 基于几何的技术 .....	22
9.2. 基于图标的技术 .....	23
9.3. Echarts 技术 .....	23
9.4. 多维数据支持和丰富视觉编码 .....	24
10. 数据库技术 .....	24
10.1. 可信任的数据库 .....	24
10.2. 数据库表关系 .....	25
10.3. 数据库表详细设计 .....	26
11. 系统实现 .....	28
11.1. 系统用户端实现 .....	28
11.1.1. APP 概况 .....	28
11.1.2. 关注内容 .....	29
11.1.3. 数据分析 .....	30
11.1.4. 评论详情 .....	32
11.2. 系统管理员端实现 .....	34
11.2.1. 用户管理 .....	34
11.2.2. 分词管理 .....	34
11.2.3. APP 分类管理 .....	34
11.2.4. APP 列表显示管理 .....	35
11.2.5. 评论列表管理 .....	35

## 1. 功能模块

### 1.1. 功能模块架构



图 1.1 功能模块架构

## 1.2. 功能模块设计

### 1.2.1. 用户端

#### ● APP 概况模块

本模块主要是对 APP 的一个总体概述，包括应用描述、基本信息两个方面。采用柱状图和折线图相结合的方式将评论量与下载量在同一图表中呈现，用户可对时间进行选择。

#### ● 关注内容模块

本模块根据每个用户的关注内容个性化设计，用户可以查看自己关注的 APP 类别以及相应的 APP，点击相应 APP 卡片即可查看该 APP 的相关信息。

#### ● 产品情感分析模块

在该模块中，利用模式匹配和机器学习相结合的算法将评论自动分为好评和差评，用户可方便地查看产品特征的好评差评详情及其百分比，有助于用户更加直观地了解产品特征的用户情感变化。

#### ● 词云图展示模块

以三维词云图的方式展现一段时间内热词的分布情况、展现每一个分词词频数随时间的变化趋势、展现一天中不同词的词频比较。

#### ● 区域分布展示模块

在该模块中，用户可以查看自己所关注的 APP 在不同国家不同地区的详细信息，也可以通过颜色的对比清楚地比较 APP 在不同地区的使用差异。

#### ● 自定义条件筛选模块

在该模块中，用户可以根据所需按时间段、按星级、按 APP 名称、按相关关键词、按平台名称对 APP 评论进行查询。

- **评论展示模块**

将评论内容、评论星级、评论时间等评论详情以表格的形式呈现出来，且采用分页技术，加快查询次数，增加用户体验。

- **APP 管理模块**

在该模块中，用户可以选择自己关注的 APP 及类别；用户可以将 APP 进行分类，更好地与其他同类 APP 进行比较。

- **个人信息管理模块**

在该模块中用户可以选择自己的部门、对自己的相关信息进行更改或处理相关邮件。

- **分词管理模块**

分词管理模块将为用户提供个性化服务为出发点，在管理员为其设定的特定词库基础上，可根据所需选择性添加分词，在下次查询时优先分析展示。

### 1.2.2. 管理员端

- **列表展示模块**

展示从数据库中获得的用户信息，包括用户名、账号、分组等重要信息，管理员可为不同的用户分组，进而与相关的权限进行绑定。

- **权限管理模块**

在该模块中，管理员可以为不同的组分配可以查看的 APP 的相关信息，进而为不同的组提供个性化的服务。

- **本地导入模块**

通过点击相应按钮，选择本地已经整理好的 Excel 内的评论内容，导入到数据库，并可以通过评论展示模块显示出来。

- **自动导入模块**

管理员通过配置获取地址、用户名和密码，即可自动从网络平台获取数据，并自动对获取评论进行分词并分类，分类结果保存在数据库中。

- **评论展示模块**

该模块是导入成功与否的验证环节，管理员无需登录数据库查看是否导入成功，在后台管理员界面就可以查看评论的详细信息。

- **APP 详情模块**

在该模块中，呈现的是 APP 在各大应用商店各个时间段的总体情况。

- **分词模块**

在该模块中，用户可以为不同的 APP 分配不同的分词词库，并可随时根据所需增加分词，为不同的分词规定优先级。

- **分类管理模块**

该模块是分词管理的基础，将不同类型的 APP 进行分类管理，也可以通过点击相关按钮进行增加或删除 APP。

## 2. 爬虫技术

### 2.1. 爬虫介绍

由于期望获得来自各大平台的 APP 评论以及其他相关数据，为了减少管理者的工作，通过爬虫技术自动获取评论数据并结构化地存入数据库中。

已覆盖的平台有 APP Store、豌豆荚、百度手机助手、vivo 应用商店、应用宝、小米应用商店、360 手机助手、oppo 应用商店、魅族应用商店。

针对于某一 APP，管理者只需配置其发布的平台，就可实现将该 APP 位于该平台的数据进行自动导入。页面解析

#### 2.1.1. 使用语言

我们选择 Python 作为爬虫语言。该语言简洁方便、对于文本及字符串的处理具有优势、同时具有多种成熟爬虫框架可供选择。

#### 2.1.2. 数据来源分析

评论的数据来源主要可分为官方 API 评论数据接口、静态页面中的数据、应用 AJAX 技术实现异步加载的动态数据。

#### 2.1.3. 页面解析工具

应用 Python 中的 Beautiful Soup 库从 HTML 或 XML 文件中提取结构性目标数据。该工具主要功能为将 HTML 的标签文件解析成树形结构，然后方便地获取到指定标签的对应属性。因此对于静态页面，只要标定评论数据位于页面中的位置属性，就能方便地解析获取评论文本。

### 2.2. 爬虫框架

采取 Python 中最为流行的爬虫框架 Scrapy。该框架中的组件包括引擎（Scrapy Engine）、调度器（Scheduler）、下载器（Downloader）、爬虫器（Spiders）、数据管道（Item Pipeline）。这些组件的互相协调具有以下优点：

- 实现并行爬取数据，提高爬取速度；
- 通过分布式爬虫避免应用平台的反爬虫机制，降低本地 IP 被封的几率；
- 通过 PhantomJS 抓取 AJAX 异步加载的请求地址，实现 AJAX 异步加载的评论数据的获取；
- 拥有完整错误处理机制，当任务出错中断可进行自动恢复，避免数据遗漏出错。

### 2.3. 爬虫结构及部署

将爬虫部署在云端 Linux 服务器中，通过 Crontab 定时任务在每日固定时间进行爬取。并将获得数据处理后自动存入数据库中，省去了工作人员手动获取评论数据的工作。

当出现较大故障爬虫无法正常工作，会进行报警提醒，管理员可进行察看排错。

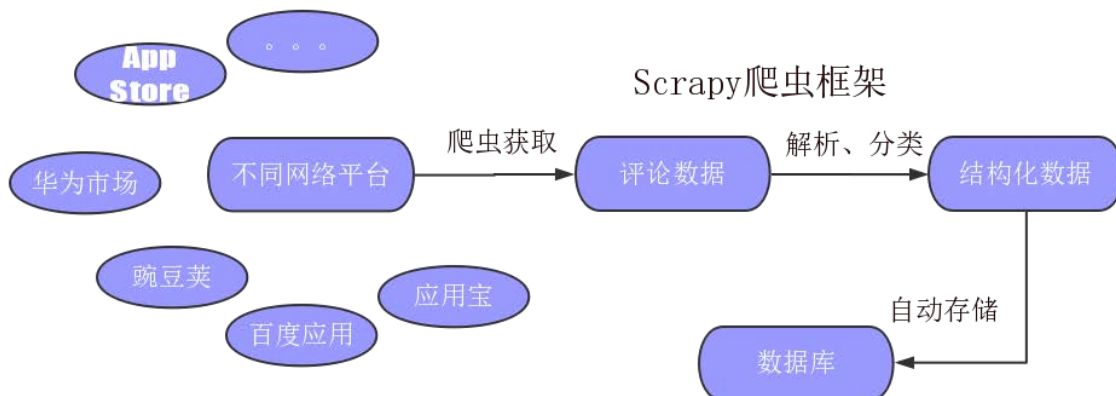


图 2.1 爬虫结构

## 2.4. 爬取数据展示

```

桌面 -- bash -- 97x42
~/Desktop -- bash

Last login: Wed May 10 18:28:10 on ttys000
You have new mail.
Age:~ apple$ cd Desktop/
Age:Desktop apple$ python claw_comment.py
开始抓取APP:meiyan的从2017-02-01到2017-02-08的数据 *** 数据表:meiyan_comment
//anaconda/lib/python3.5/site-packages/pymysql/cursors.py:323: Warning: (1265, "Data truncated fo
r column 'date' at row 1")
self._do_get_result()

-----
用户 : 粉色味蕾的悲伤 | 评论 : | 平台 : 应用宝
-----
用户 : sunny欣 | 评论 : 。 | 平台 : 豌豆荚
-----
用户 : 182****9012 | 评论 : 可以 | 平台 : 豌豆荚
-----
用户 : 177****4373 | 评论 : 可以的 | 平台 : 豌豆荚
-----
用户 : 婷 | 评论 : 好 | 平台 : 豌豆荚
-----
用户 : 一曲离殇、梨花雨凉 | 评论 : 不错 | 平台 : 豌豆荚
//anaconda/lib/python3.5/site-packages/pymysql/cursors.py:323: Warning: (1366, "Incorrect string
value: '\\xF0\\x9F\\x91\\x89\\xE9\\xA...' for column 'name' at row 1")
self._do_get_result()

-----
用户 : 随遇而安 | 评论 : 挺不错的, 赞一个, 再也不用担心颜值低害怕拍照了 | 平台 : AppStore
-----
用户 : 随遇而安 | 评论 : 挺不错的, 赞一个, 再也不用担心颜值低害怕拍照了 | 平台 : AppStore
-----
用户 : 女人可以哭 不可以输 | 评论 : 很好用的 | 平台 : 应用宝
-----
用户 : 941899998 | 评论 : 挺挺好哒, 整个人都变得美美哒 | 平台 : 小米应用商店
-----
用户 : 130****657 | 评论 : 很好用, 非常好用 | 平台 : 百度手机助手
-----
用户 : 谁家记忆 | 评论 : 好 | 平台 : 豌豆荚
-----
用户 : 爱到荼靡 | 评论 : 很好, very good! | 平台 : 应用宝
-----
用户 : 158****0479 | 评论 : 还可以 | 平台 : 豌豆荚
-----
用户 : 133****7447 | 评论 : 好好好好 | 平台 : 豌豆荚
-----

```

图 2.2 「美颜相机」评论爬取示例

表 2.1 「美颜相机」三月评论数平台分布

平台	APP Store	百度手机助手	小米应用	OPPO 应用商店	Vivo 应用商店	...
三月评论数量	22016	16053	10696	16676	8291	...



表 2.2 「Instagram」不同地区评论分布

地区	中国	美国	韩国	巴西	法国	...
一周评论数量	959	1432	160	295	152	...

### 3. 算法

#### 3.1. 分词算法

##### 3.1.1. 算法介绍

###### ● 基于 Trie 树结构实现词典扫描并生成有向无环图 (DAG)

本系统使用的结巴分词器自带了一个叫做 dict.txt 的词典，里面有 2 万多条词，包含了词条出现的次数和词性。Trie 树结构实现的词图扫描，就是把这 2 万多条词语，放到一个 Trie 树中，而 Trie 树是有名的前缀树，也就是说一个词语的前面几个字一样，就表示他们具有相同的前缀，就可以使用 Trie 树来存储，具有查找速度快的优势。

DAG 有向无环图，就是后一句的生成句子中汉字所有可能成词情况所构成的有向无环图，即就是给定一个待分词的句子，对这个句子进行生成有向无环图。切分步骤如下所示：

- 1) 根据 dict.txt 生成 Trie 树
- 2) 对待分词句子，根据 dict.txt 生成的 Trie 树，生成 DAG，实际上通俗的说，就是对待分词句子，根据给定的词典进行查词典操作，生成几种可能的句子切分。

例如：{0: [1, 2, 3]} 这样一个简单的 DAG，就是表示 0 位置开始，在 1, 2, 3 位置都是词，就是说 0~1, 0~2, 0~3 这三个起始位置之间的字符，在 dict.txt 中是词语。

###### ● 采用动态规划查找最大概率路径并找出最大切分组合

字典在生成 Trie 树的同时，也把每个词的出现次数转换为了频率。对于频率和概率，按照定义，频率其实也是一个 0~1 之间的小数，是事件出现的次数/实验中的总次数，因此在试验次数足够大的情况下，频率约等于概率，或者说频率的极限就是概率。

动态规划中，先查找待分词句子中已经切分好的词语，对该词语查找该词语出现的频率(次数/总数)，如果没有该词(既然是基于词典查找，应该是有的)，就把词典中出现频率最小的那个词语的频率作为该词的频率，也就是说  $P(\text{某词语}) = \text{FREQ.get}(\text{'某词语'}, \text{min\_freq})$ ，然后根据动态规划查找最大概率路径的方法，对句子从右往左反向计算最大概率，因为汉语句子的重心经常落在后面，就是落在右边，因为通常情况下形容词太多，后面的才是主干。因此，从右往左计算，正确率要高于从左往右计算，这个类似于逆向最大匹配)， $P(\text{NodeN}) = 1.0$ ， $P(\text{NodeN}-1) = P(\text{NodeN}) * \text{Max}(P(\text{倒数第一个词})) \dots$ 依次类推，最后得到最大概率路径，得到最大概率的切分组合。

###### ● 使用 HMM 模型和 Viterbi 算法对未登陆词进行识别

给定一个待分词的句子，利用 HMM 模型可以实现为其分词。中文词汇按照 BEMS 四个状态来标记。对 HMM(BEMS) 四种状态的模型来说，就是为了找到一个最佳的 BEMS 序列，这个就需要使用 Viterbi 算法来得到这个最佳的隐藏状态序列。

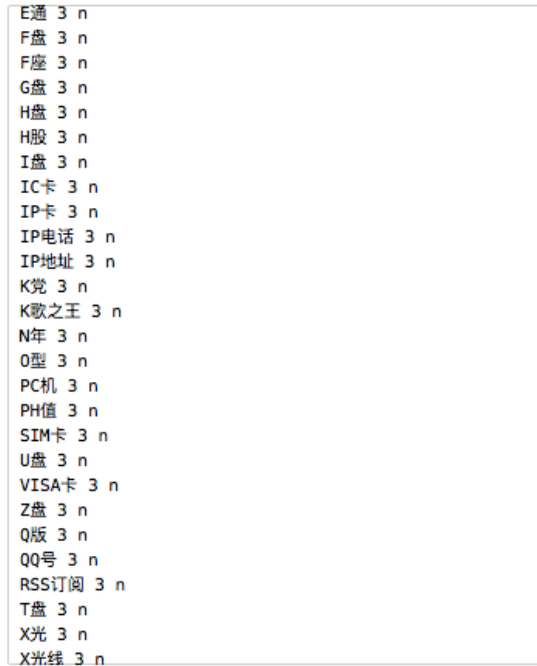
通过训练得到的概率表和 Viterbi 算法，就可以得到一个概率最大的 BEMS 序列，按照 B 打头，E 结尾的方式，对待分词的句子重新组合，就得到了分词结果。比如对待分词的句子‘全世界都在学中国话’得到一个 BEMS 序列[S, B, E, S, S, S, B, B,



E, S], 通过把连续的 BE 凑合到一起得到一个词, 单独的 S 放单, 就得到一个分词结果了: 上面的 BE 位置和句子中单个汉字的位置一一对应, 得到全/S 世界/BE 都/S 在/S 学/S 中国/BE 话/S 从而将句子切分为词语。

### 3.1.2. 分词过程

- 1) 加载字典 dict.txt (该词典来源于中科院 NLPID 分词词库), 生成 Trie 树;



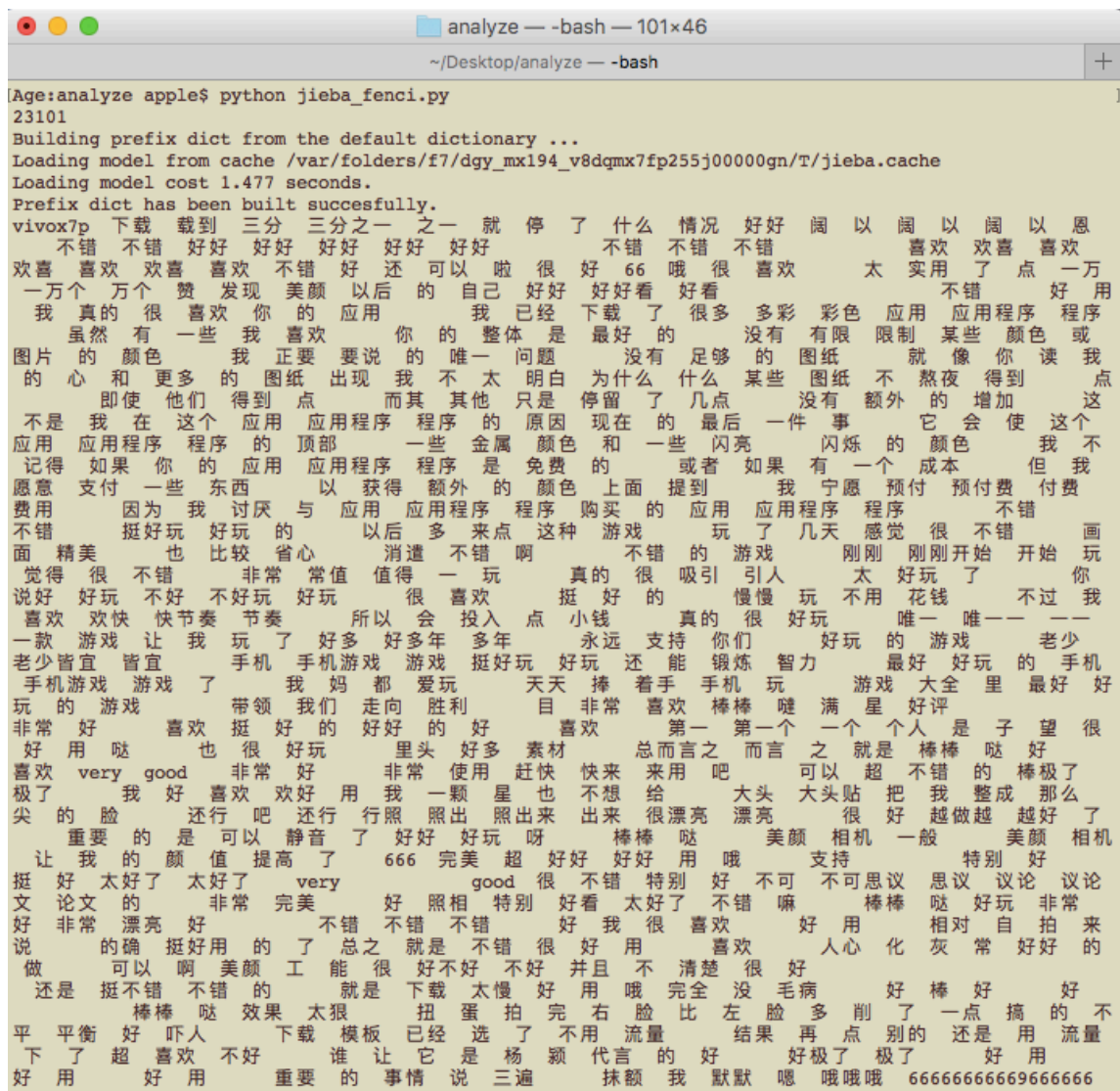
```

E通 3 n
F盘 3 n
F座 3 n
G盘 3 n
H盘 3 n
H股 3 n
I盘 3 n
IC卡 3 n
IP卡 3 n
IP电话 3 n
IP地址 3 n
K党 3 n
K歌之王 3 n
N年 3 n
O型 3 n
PC机 3 n
PH值 3 n
SIM卡 3 n
U盘 3 n
VISA卡 3 n
Z盘 3 n
Q版 3 n
QQ号 3 n
RSS订阅 3 n
T盘 3 n
X光 3 n
X光线 3 n
    
```

dict.txt

图 3.1 中科院 NLPID 分词基础词典

- 2) 给定待分词的句子, 使用正则获取连续的中文字符和英文字符, 切分成短语列表, 对每个短语使用 DAG (查字典) 和动态规划, 得到最大概率路径, 对 DAG 中那些没有在字典中查到的字, 组合成一个新的片段短语, 使用 HMM 模型进行分词, 也就是识别新词, 即识别字典外的新词;
- 3) 使用 python 的 yield 语法生成一个词语生成器, 逐词语返回。



```

analyze --bash 101x46
~/Desktop/analyze --bash
Age:analyze apple$ python jieba_fenci.py
23101
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/f7/dgy_mx194_v8dgm7fp255j00000gn/T/jieba.cache
Loading model cost 1.477 seconds.
Prefix dict has been built successfully.
vivox7p 下载 载到 三分 三分之一 之一 就 停 了 什么 情况 好好 阅 以 阅 以 阅 以 恩
不错 不错 好好 好好 好好 好好 好好 好好 不错 不错 不错 喜欢 欢喜 喜欢
欢喜 喜欢 欢喜 喜欢 不错 好 还 可以 啦 很 好 66 哦 很 喜欢 太 实用 了 点 一万
一万个 万个 赞 发现 美颜 以后 的 自己 好好 好好 好看 好看 不错 好 用
我 真的 很 喜欢 你的 应用 我 已经 下载 了 很多 多彩 彩色 应用 应用程序 程序
虽然 有 一些 我 喜欢 你的 整体 是 最好的 没有 有限 限制 某些 颜色 或
图片 的 颜色 我 正要 要说的 唯一 问题 没有 足够 的 图纸 就 像 你 读 我
的 心 和 更多 的 图纸 出现 我 不 太 明白 为什么 什么 某些 图纸 不 熬夜 得到 点
即使 他们 得到 点 而其 其他 只是 停留 了 几点 没有 额外 的 增加 这
不是 我 在 这个 应用 应用程序 程序 的 原因 现在 的 最后 一件 事 它 会 使 这个
应用 应用程序 程序 的 顶部 一些 金属 颜色 和 一些 闪亮 闪亮 的 颜色 我 不
记得 如果 你的 应用 应用程序 程序 是 免费 的 或者 如果 有一个 成本 但 我
愿意 支付 一些 东西 以 获得 额外 的 颜色 上面 提到 我 宁愿 预付 预付费 付费
费用 因为 我 讨厌 的 应用 应用程序 程序 购买 的 应用 应用程序 程序 不错
不错 挺好 好玩 好玩 以后 多 来点 这种 游戏 玩 了 几天 感觉 很 不错 画
面 精美 也 比较 省心 消遣 不错 啊 不错 的 游戏 刚刚 刚刚开始 开始 玩
觉得 很 不错 非常 常值 值得 一 玩 真的 很 吸引 引人 太 好玩 了 不过 你
说好 好玩 不好 好玩 好玩 很好 挺好 的 慢慢 玩 不用 花钱 不过 我
喜欢 欢快 快节奏 节奏 所以 会 投入 点 小钱 真的 很好 玩 唯一 唯一 唯一
一款 游戏 让 我 玩 了 好多 好多年 多年 永远 支持 你们 好玩 的 游戏 老少
老少 皆宜 皆宜 手机 手机游戏 游戏 挺好 玩 好玩 还能 锻炼 智力 最好 好玩 的 手机
手机游戏 游戏 了 我 妈 都 爱玩 天天 捧 着手 手机 玩 游戏 大全 里 最好 好
玩 的 游戏 带领 我们 走向 胜利 目 非常 喜欢 棒棒 哒 满 星 好评
非常好 喜欢 挺好 的 好好的 好 喜欢 第一 第一个 一个 人 是 子 望 很
好用 哒 也 很好 玩 里头 好多 素材 总 而 之 而 之 就是 棒棒 哒 好
喜欢 very good 非常好 非常 使用 赶快 快来 来用 吧 可以 超 不错 的 棒极了
极了 的 脸 还行 吧 还行 行照 照出 照出来 出来 很漂亮 很漂亮 很好 越做越 越好了
尖 的 重要 的 是 可以 静音 了 好好 好玩 呀 棒棒 哒 美颜 相机 一般 美颜 相机
让 我的 颜值 提高 了 666 完美 超 好好 好好 用 哦 支持 特别 好
挺 好 太好了 太好了 very good 很好 不错 特别 好 不可 思议 思议 讨论 讨论
文 论文 的 非常 完美 好 照相 特别 好看 太好了 不错 嘛 棒棒 哒 好玩 非常
好 非常 漂亮 好 不错 不错 不错 好 我 很 喜欢 好用 相对 自拍 来
说 的确 挺好 用的 了 总之 就是 不错 很好 用 喜欢 人心 化 灰 常 好好的
做 可以 啊 美颜 工 能 很 好 不好 不好 并且 不 清楚 很好
还是 挺不错 不错 的 就是 下载 太慢 好用 哦 完全 没 毛病 好 棒 好 好
平 平衡 好 吓人 下载 模板 已经 选 了 不用 流量 结果 再 点 别的 还是 用 流量
下 了 超 喜欢 不好 谁 让 它是 杨 颖 代言 的 好 好极了 极了 好 用
好用 好用 重要 的 事情 说 三遍 抹额 我 默默 嗯 哦哦哦 6666666666666666

```

图 3.2 分词结果

### 3.2. 关键词提取算法

#### 3.2.1. TF-IDF 算法介绍

我们选取 TF-IDF 算法作为关键词选取的主要算法，该算法在搜索引擎等实际应用中广泛使用，主要用以评估一个字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

#### 3.2.2. TF-IDF 算法使用

该算法以 TF 和 IDF 的乘积作为特征空间坐标系的取值测度，并用它完成对权值 TF 的调整，调整权值的目的在于突出重要单词，抑制次要单词，通过该方式，我们即可获得用户希望获得的关键词。TF-IDF 计算公式如下所示：

$$TF-IDF = TF * IDF$$

在一个文本中出现很多次的单词，在另一个同类文本中出现次数也会很多，反之亦然。所以如果特征空间坐标系取 TF 词频作为测度，就可以体现同类文本的特点，所以

引入了词频 TF 的概念。TF (Term Frequency)，意为词频，用于计算该次描述文档内容的能力，词频计算公式如下所示：

$$TF = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

Viision—APP 评论数据分析系统将一段时间内的所有评论内容放在一个文件中，因评论内容涵盖各个应用商店，可保证评论内容足够多，进而可类比于公式中的文章。

TF-IDF 算法认为一个单词出现的文本频率越小，它区别不同类别的能力就越大，所以引入了逆文本频度 IDF 的概念。IDF (Inverse Document Frequency)，意为“逆文档频率”，用于计算该词区分文档的能力。就是在词频的基础上，要对每个词分配一个“重要性”权重。最常见的词（如：“的”、“是”、“在”）给予最小的权重，较常见的词（如：“内容”）给予较小的权重，较少见的词（如：“补光”、“妆容”）给予较大的权重。当一个词在这篇文档中出现的频率越高，同时在其他文档中出现的次数越少，则表明该词对于表示这篇文档的区分能力越强，所以其权重值就应该越大。IDF 计算公式如下所示：

$$IDF = \log \left( \frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1} \right)$$

### 3.2.3. TF-IDF 优缺点分析

TF-IDF 算法的优点是：简单快速，易于理解与操作，结果比较符合实际情况。

TF-IDF 算法的缺点是：

- 1) 单纯以“词频”和“逆文档频率”衡量一个词的重要性，对于本项目 Viision—APP 评论数据分析系统来说不够全面，会出现一些重要词因出现次数并不多而被忽略的情形。
- 2) 引入 IDF 调整权值的目的在于突出重要单词，抑制次要单词。实际上，IDF 是一种试图抑制噪音的加权，并且单纯地认为文本频数小的单词就越重要，文本频数大的单词就越无用，显然这并不是完全正确的。
- 3) 在 TF-IDF 算法中并没有体现出单词的位置信息，特征词在不同的标记符中对文章内容的反映程度不同，其权重的计算方法也应不同，出现位置靠前的词与出现位置靠后的词，都被视为重要性相同，这是不完全正确的。

### 3.2.4. TF-IDF 算法改进措施

针对以上 TF-IDF 算法存在的缺点，我们做出以下改进措施：即当用户添加具体分词时，可以对这些词赋一定的权重。这样既可以帮助该关键词的提取，又不影响该关键词相对于其他词在总体评论中的权重。提取示例如下图所示。

```

Age:analyze apple$ python jieba_tfidf.py
read lens : 23101
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/f7/dgy_mx194_v8dgm7fp255j00000gn/T/jieba.cache
Loading model cost 1.358 seconds.
Prefix dict has been built succesfully.
好好好好 0.700659
不错 0.330569
美颜 0.279909
棒棒 0.219006
好用 0.212560
喜欢 0.201925
太太太太 0.193077
非常 0.169320
好玩 0.162004
相机 0.149864
美美 0.124530
软件 0.121457
好好 0.116516
下载 0.084941
太好了 0.084011
6666 0.080538
太棒了 0.075356
特别 0.069689
挺好用 0.069159
很棒 0.067025
真的 0.065602
棒棒棒棒 0.065426
赞赞 0.064359
好评 0.062339
超级 0.060748
哈哈 0.060252
好看 0.057727
颜值 0.054403
效果 0.050038
自拍 0.049120
拍照 0.048767
照片 0.042899
玩玩 0.041405
太好 0.039990
推荐 0.036538
大家 0.036079
  
```

图 3.3 评论关键词提取

### 3.3. 机器学习算法

#### 3.3.1. 基于朴素贝叶斯的评论分类算法

在实际应用中，系统往往需要对评论按照不同的要求进行分类，这时通过一般的模式匹配的方法很难完成区分包含较多属性的类别的任务。而通过使用有监督学习的朴素贝叶斯二分类器可区分包含多重属性的评论类别。该方法是在分布独立这个假设成立的情况下实现，而在该文本数据中，分布独立的假设基本成立。该方法过程简单速度快，分类效果好。

设各个单元之间两两独立。设训练样本集分为 $k$ 类，记为 $C = \{C_1, C_2, \dots, C_k\}$ ，则每个类 $C_i$ 的先验概率为 $P(C_i)$ ，其值为 $C_i$ 类的样本数除以训练集总样本数 $n$ 。对于新样本 $d$ ，其属于 $C_i$ 类的条件概率是 $P(d|C_i)$ ， $C_i$ 的后验概率为 $P(C_i|d)$ 。

$$P(C_i | d) = \frac{P(d | C_i)P(C_i)}{P(d)} \propto P(d | C_i)P(C_i)$$

#### 3.3.2. 评论分类流程

评论的检测中，首先通过停用词库、情感词库和后缀词库对数据集进行预处理；然后将每一句评论使用 TF-IDF 算法进行特征的选择和提取，并转化为可计算的 TF-IDF 向量。将训练集使用朴素贝叶斯分类器进行训练建模，训练集的选择与标记安装分类需求不同而标记不同。最后将待分类评论用生的模型进行分类。流程示意如下图所示。

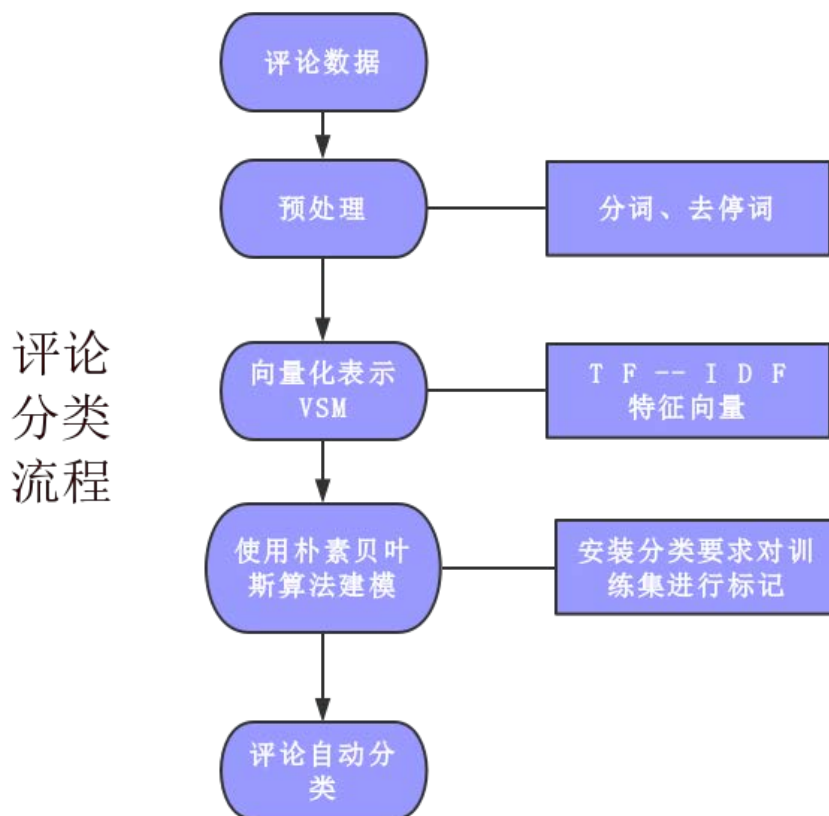


图 3.4 朴素贝叶斯评论分类

## 4. 垃圾评论过滤技术

### 4.1.1. 垃圾评论分类算法

通过使用有监督学习的朴素贝叶斯二分类器可区分有用评论与垃圾评论，进而过滤垃圾评论。内容型垃圾评论的检测中，首先通过停用词库、情感词库和后缀词库对数据集进行预处理。

然后将每一句评论使用 TF-IDF 算法进行特征的选择和提取，并使用向量空间表示模型（VSM）将评论转化为可计算的 TF-IDF 向量，向量的每一分量即为计算权重。假定有以下一句评论：

就想只是想注册下用户，搞不懂为何注册不了，还一直显示手机错误。

通过 TF-IDF 算法可以提取出该句评论的几个特征：

$$M = \{\text{注册、用户、搞不懂、手机、显示、错误}\}$$

每个特征具有 TF-IDF 权重，其权重大小来自于上述 TF-IDF 求特征：

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$$

最后评论就可以表示成一个特征向量，记为：

$$\vec{D} = \{m_1f_1, m_2f_2, m_3f_3, m_4f_4, m_5f_5, m_6f_6\}$$

将训练集使用朴素贝叶斯分类器进行训练建模。对于待检测的评论，将其置于模型中计算出分类概率，若为垃圾评论的概率大于  $1/2$ ，标记为垃圾评论。分类器测试结果如下图所示：期中共测试数据 8420 条，正确率达 84%。



```

桌面 — -bash — 69x17
~/Desktop — -bash
[Age:Desktop apple$ python classify_filter.py
正在准备训练和测试数据, 请稍后...
准备训练和测试数据准备完毕, 下一步...
计算TF-IDF, 提取关键/特征词中, 请稍后...
使用朴素贝叶斯训练中...
训练完毕, 写入模型...
加载模型中...
正在使用测试数据验证模型效果...
总共测试文本量: 8420, 预测正确的类别量: 7133, 朴素贝叶斯分类器准确度:
84.098%
Age:Desktop apple$
    
```

图 4.1 垃圾评论过滤模型测试

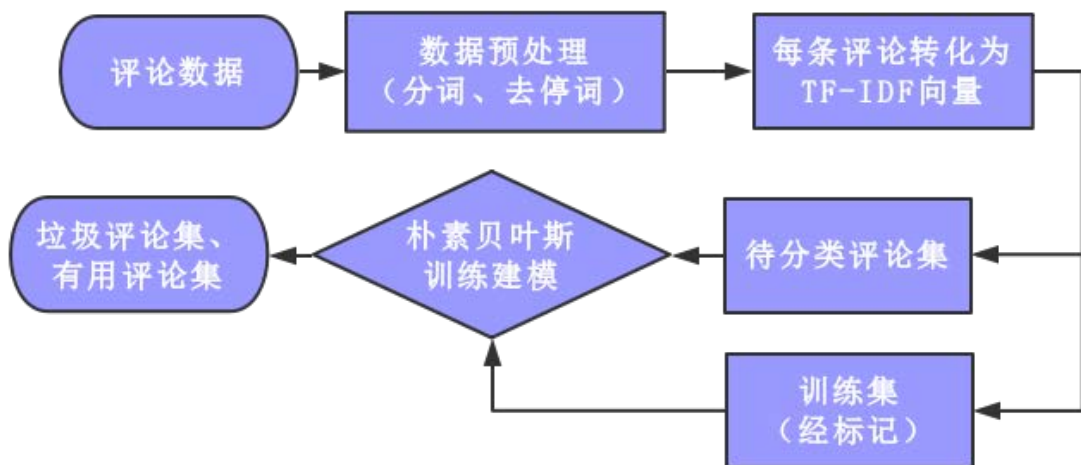


图 4.2 垃圾评论过滤框架

#### 4.1.2. 垃圾评论训练集的设计及更新

由于垃圾评论筛选的模型是基于训练集进行训练得出的，因此训练集需覆盖类型较全的垃圾评论。来源可分为网络中权威的垃圾评论集经过整理得出的集合和用户从评论中自主筛选标记的垃圾评论。以此方式得出的训练集具有较好的代表性，保障具有较好的垃圾评论过滤结果。

同时由于垃圾评论的类型样式变化迅速，因此在一段时间后将当下流行的垃圾评论类型集加入到训练集中也是必要的。垃圾评论示例如下表所示。



[illegible]

垃圾评论.txt

图 4.3 垃圾评论库

## 5. 关键词提取技术

### 5.1.1. 关键词分类

由于不同的部门对于评论信息有不同方面的需求,因此针对不同部门用户进行关键词展示时可采用不同的功能类词集,这些文字集大部分可通过网络现有的词集获取。这里我们针对「摄影工具」类对不同的部门整理关键词。具体关键词词库示例如下表所示:

表 5.1 关键词词库分类

部门	类别	具体关键词
营销策划部门	活动类关键词	购买商品打折优惠抢购余额购物车交易理财钱包免费会员会员卡奖品骗子支付红包奖励福利消费付费价格特价业务抽奖活动

		贵族赠送中秋春节五一代言玩家模式玩法券券代言女神榜首
开发部	功能类关键词	美颜软件下载推荐功能自拍建议发现使用评论微信分享关注滤镜使用设计电话推荐评价表情美化自拍界面保存视频使用效果神器大头贴化妆朋友圈游戏整容图片皮肤相片一键磨皮动画补光妆容装饰素材照相机风景手绘漫画网红脸型水印静音
产品部	问题类关键词	下载更新手机广告垃圾骗子问题评价内存版本黑屏无聊死机运行封号注册错误登录乱反馈商家客服像素习惯时间画面团队内容太假权限差评信息总会速度不卡卡真卡
全部部门	情感程度关键词	好真好很好挺好有点超棒挺棒棒极了真棒不错呀不错不错呀给力好玩

### 5.1.2. 关键词提取

对于不同部门的用户，加载各自所属的关键词备选表，使用 TF-IDF 算法提取排名前 8 位词，并按照权重进行排序。

这里我们将应用「美颜相机」于近一个月在各平台上的全部评论数据进行关键词提取，针对各功能类提取关键词获得的详细结果如下表：（其中权重代表该词在评论中的重要程度，详见 TF-IDF 算法使用）

表 5.2 功能类关键词筛选

关键词	权重
美颜	0.279267
软件	0.122494
下载	0.083518
自拍	0.052284
效果	0.050256
推荐	0.035647
神器	0.029505
大头贴	0.027193

表 5.3 问题类关键词筛选

关键词	权重
垃圾	0.173095

手机	0.114964
广告	0.070160
像素	0.040791
版本	0.035307
内存	0.026239
习惯	0.025426
问题	0.021667

### 5.1.3. 个性化分词添加

由于默认提供的关键词库存在一定的局限性，同时每一阶段会出现网络新词以及新的需求，按照原有词库进行关键词提取并不能满足用户需求。因此对用户提供了自定义添加分词的入口，用户可根据自己的需求个性化地添加新词至词库中。经重新进行关键词提取后，将优先展示。

## 6. 评论情感分析技术

### 6.1. 机器学习算法

#### 6.1.1. 情感分析算法

通过使用有监督学习的朴素贝叶斯二分类器可区分评论情感的极性，判断评论情感的褒贬性。首先进行评论的预处理；然后将每一句评论使用 TF-IDF 算法进行特征的选择和提取，并转化为可计算的 TF-IDF 向量。将训练集使用朴素贝叶斯分类器进行训练建模。对于待检测的评论，将其置于模型中计算出分类概率，计算评论所属极性的类别。

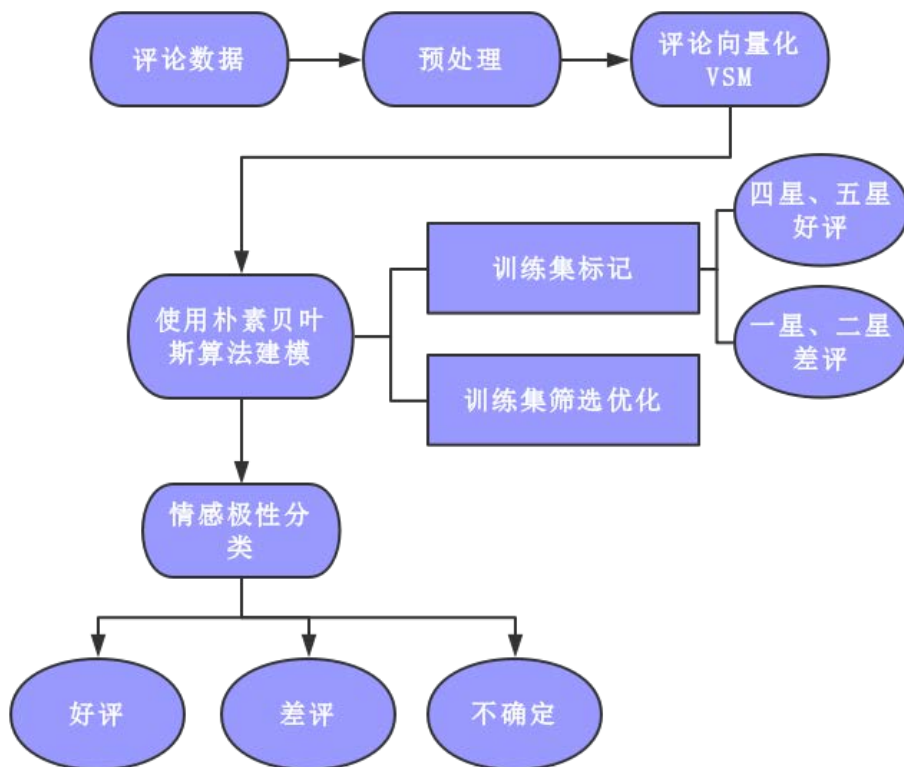


图 6.1 基于朴素贝叶斯的评论情感分析框架

### 6.1.2. 评论不同极性训练集的设计及更新

我们将评论的情感极性分成褒义、贬义和中性三种极性。针对极性明显的褒义和贬义的评论训练集，我们选取评论原始数据中的星级数作为参照标准进行标记，即 4 星 5 星标记成褒义，1 星 2 星标记成贬义（共 5 星）。

为了避免评论用户乱评星级导致星级数与情感极性不符的情况出现，我们采取多次选取不同数据集进行测试最终选取最优训练模型的方式来保证评论星级与情感极性的匹配性。

## 6.2. 模式匹配算法

上述基于统计的自动分类对训练语料的数量和质量的要求高，当一些类目没有充足的训练语料或语料的质量不高时，情感极性分类效果不好，可以采用基于规则的分类来做进一步优化。

## 6.3. 混合模式

机器学习分类方式具有训练简单，分类精度高的优点。其缺点主要是对训练集数据的数量和质量有严格的要求。当训练集数量不全面或代表性不强时，将严重影响分类效果。因此对于极性明显的评论能较好区分，而对于态度不鲜明的评论该方式极性分类效果较差。

而基于模式匹配的方式采用对评论进行情感特征词的匹配的方式进行情感分析。因此对于评论极性不明显的方式采用该方法比较有效。

综合以上两种方法优缺点，我们采取两种方式混合的模式进行评论的情感分析，即主要采用机器学习方式进行评论情感极性判定，对于不适用该方式进行分类的评论数据，通过模式匹配的方式进行辅助情感判断，以此提高评论情感极性判断的准确性。

表 6.1 特征属性「像素」的情感分析

好评	差评
很好，真心的照相好看， <b>像素</b> 很好	<b>像素</b> 好差呀，好烦啊，没一个好用的？
<b>像素</b> 超好，赞赞哒???	<b>像素</b> 太差，其它都可以
拍的照片比我原来的手机自拍 <b>像素</b> 好多了，超赞！	拍出的照片儿太显黑..... <b>像素</b> 不好
不错比苹果本身 <b>像素</b> 好多了，颜值提高了很多呢？	这什么鬼？更新后 <b>像素</b> 忒不好了、太失望、而且超级超级不清楚，都想要换 app 的节奏
棒棒哒！此相机有高清的 <b>像素</b> ，萌萌的大头贴！很好用哦。	说实话对于安卓 <b>像素</b> 低了些介意的慎下
好用 太好用了 一定要下载这个美颜相机 <b>像素</b> 高 质量好 我用了快1年了	越更新越不好 <b>像素</b> 变模糊了照出来的太难看没有以前的好看了喜欢以前的那个版本
美颜相机拍的， <b>像素</b> 很好，效果也好，上面的大头贴功能也特别新颖。	起其他美颜相机 <b>像素</b> 低了好多拍出来的照片有点粗糙什么时候能改善还有就是耗电太厉害

## 7. 软件技术

### 7.1. Laravel 框架

#### 7.1.1. 模型—视图—控制器

Laravel 是 Model-View-Controller (MVC) 架构模式，它使用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。



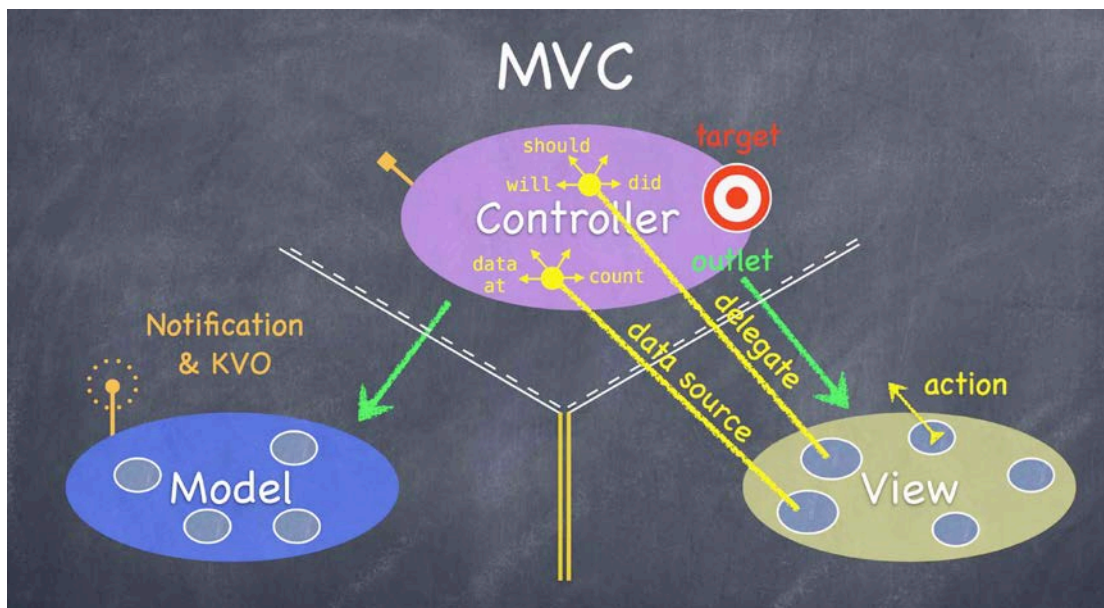


图 7.1 LaravelMVC 架构

### 7.1.2. 响应流程

当访问一个 laravel 应用程序时，浏览器发送一个请求，由 Web 服务器接收并传递到 laravel 的路由引擎。该 laravel 路由器接收到请求后，根据配置重定向到相应的控制器类的方法。

然后由控制器类接管。在某些情况下，控制器将立即渲染一个视图，这是一个模板，将被转换成 HTML 并且发送回浏览器。更普遍的动态网站，控制器与模型进行交互，与数据库进行通信。调用模型后，控制器呈现最终视图 (HTML、CSS 和图像) 并返回完整的 web 页面到用户的浏览器。

Laravel 提倡模型、视图和控制器，应保持完全独立的存储单独的文件在不同的目录，这就是 laravel 的目录结构发挥作用的地方。

## 7.2. Nginx 服务器

Nginx 服务器是一款轻量级的 Web 服务器和电子邮件代理服务器，并在一个 BSD-like 协议下发行其稳定性好、功能集丰富、系统资源低、占有内存少，并发能力强，且其并发能力在同类型的网页服务器中表现突出。

## 7.3. 基于 Ajax 技术的 Web 服务架构

在传统的 Web 服务模式，用户和服务端之间是一种同步关系，服务器在处理请求的时候，用户多数时间只能等待，限制了交互性，用户体验较差。基于 Ajax 技术的 Web 服务架构为浏览器提供了与服务端异步通信的能力，可以实现页面的局部刷新而不是加载整个页面，减少了用户等待的时间，更好的满足了用户需求，使得 Web 应用程序更加人性化。

Ajax 即 “Asynchronous JavaScript and XML”，是一种创建交互式网页应用的网页开发技术。Ajax 技术实现过程是，Web 页面中的 JavaScript 脚本使用 XMLHttpRequest 对象与服务端异步通信，服务器接收请求后返回业务数据；数据通过脚本程序处理后，经过数据可视化技术更新显示在 Web 页面中。这种异步数据读取方法使 Ajax 可以自主的发起 Web 请求，与远端服务器完成必要的的数据交互，在构建 Web 页面时，



无需中断交互流程即可重新加载和动态更新，既减轻了服务器负载又加快了响应速度，缩短了用户等待的时间。

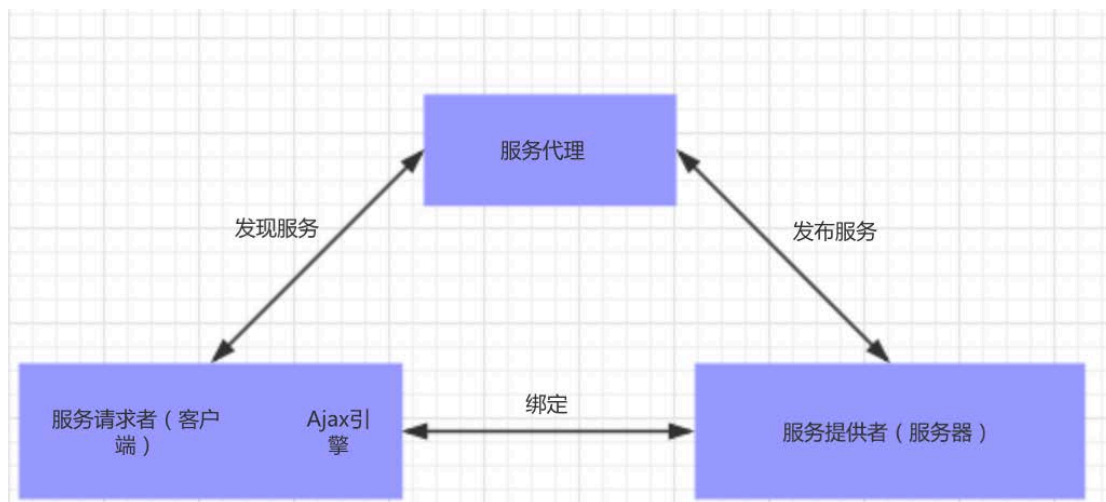


图 7.2 基于 Ajax 技术的 Web 服务架构

本项目应用此技术可以在不刷新全部页面的情况下加载 app 的数据，用户可以方便快捷的查看到 app 在不同时间段内的各种情况，以及不同筛选条件下的评论情况，提高了数据加载的速度，增加了用户体验。

## 8. 前端交互技术

### 8.1. 框架选择

Viision—APP 评论数据分析系统前端设计框架采用的是 Google 推出的一款重视跨平台体验的设计语言——Material design。

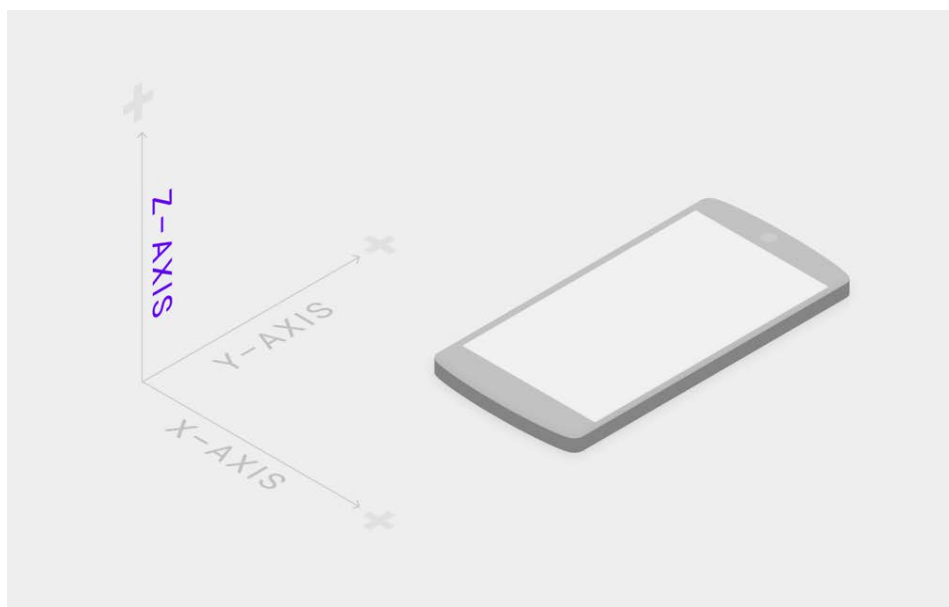


图 8.1 Material design

Material design 期望的是把物理世界的体验带进屏幕，去掉现实中的杂质和随机性，保留其最原始纯净的形态、空间关系、变化与过渡。Viision 所做的也是最直观的数据体现，沉浸式的数据分析氛围，配合 Material design 构建的虚拟世界的灵活特性，还原最贴近真实的体验，达到简洁与直观的效果。

## 8.2. 产品颜色设计

### ● 原色

Viiision—APP 评论数据分析系统应用程序屏幕和组件中最常显示的颜色为一种优雅的紫色，它是我们的原色。

同时，为了创建元素之间的对比度，我们使用较浅或较深色调的原色，增添明度和饱和度的变化，较浅和较暗色调之间的对比有助于显示表面之间的划分，例如状态栏和工具栏之间的划分。

### ● 次要颜色

辅助颜色用于重写 UI 的部分选择。它是原色紫色的互补色，一种荧光绿。它与周围的元素形成对比，突出强调。



图 8.2 网页颜色

## 8.3. 响应式布局

Material design 要求这些布局能够适应任何屏幕尺寸。为了简化适用于各种尺寸的分屏模式的应用程序，我们最先设计最小的尺寸。利用栅格化样式，确保布局之间的一致性。有利于用户在手机端进行方便地浏览。

### ● 布局摘要及详细视图内容

600dp 以下的布局可以用单级内容层次结构填充屏幕（摘要或详细内容，但不同时使用）。

超过 600dp 的布局可以在屏幕上放置两个层次的内容层次结构（摘要和详细内容）

### ● 最大屏幕宽度

1600dp 宽的布局可能会使布局增长，直到达到最大宽度。此时，栅格会执行以下操作：

- 1) 成为中心与边缘增加一致
- 2) 保持左对齐，右边距增长

### 3) 继续增长，同时显示其他内容



图 8.3 操作端界面示例

## 9. 数据可视化技术

数据可视化是利用计算机图形学的图像处理技术，将数据转换成图形或图像在屏幕上显示出来，并进行交互处理的理论、方法和技术。

### 9.1. 基于几何的技术

基于几何的可视化技术包括 Scatter plots、Landscapes、Projection Pursuit、Parallel Coordinates 等等，是以几何画法或几何投影的方式来表示数据库中的数据。平行坐标法是最早提出的以二维形式表示  $n$  维数据的可视化技术。它的基本思想是将  $n$  维数据属性空间通过  $n$  条等距离的平行轴映射到二维平面上，每一条轴线代表一个属性维，轴线上的取值范围从对应属性的最小值到最大值的均匀分布。这样，每一个数据项都可以根据其属性值用一条折线段在  $n$  条平行轴上表示出来。

利用这个技术加上 d3.js 我们可以设计出符合要求的  $x, y$  轴，图表可以跨坐标系存在，例如折、柱、散点等图可以放在直角坐标系上，也可以放在极坐标系上，甚至可以放在地理坐标系中。以及合适的范围数据和比例尺。

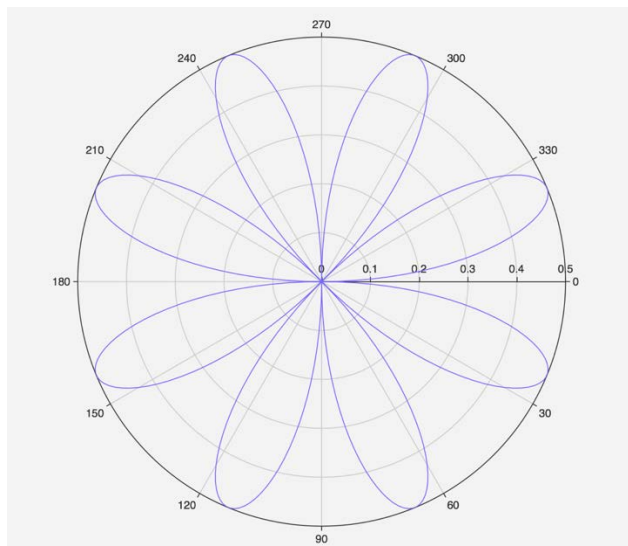


图 9.1 极坐标双数值设计

## 9.2. 基于图标的技术

基于图标技术的基本思想是用一个简单图标的各个部分来表示  $n$  维数据属性。基于图标的可视化技术包括 Chernoff-face、Shape Coding、Stick Figures 等，这种技术适用于某些维值在二维平面上具有良好展开属性的数据集

枝形图方法是其中的基本方法之一。枝形图方法首先选取多维属性中的两种属性作为基本的 X-Y 平面轴，在此平面上利用小树枝的长度或角度的不同表示出其他属性值的变化。

本项目利用多维属性在一个图表中表示出一个或多个 app 的多个属性，可以使用户在短时间有效快速的获得 app 的不同方面的信息。例如下图所示的两个数据点，它们对左边的二维属性含有相同的数据值，而右边的二维属性的数据值则不相同。

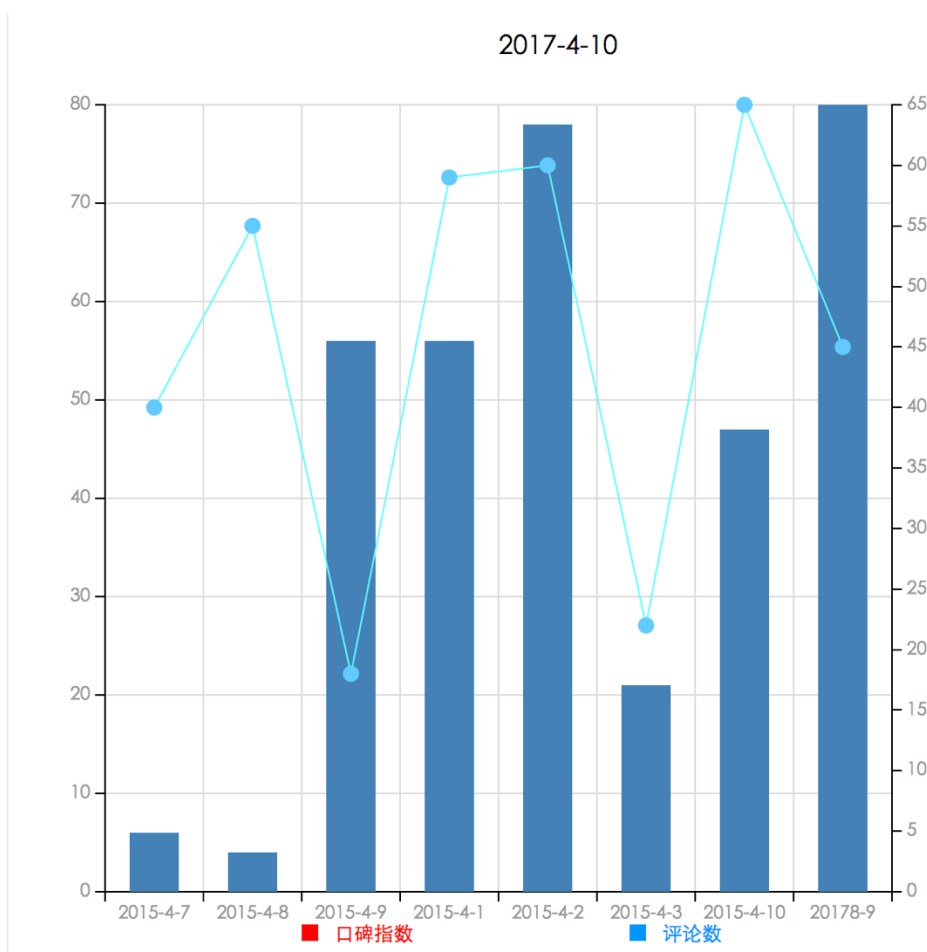


图 9.2 属性图标

## 9.3. Echarts 技术

ECharts，一个纯 Javascript 的图表库，可以流畅的运行在 PC 和移动设备上，底层依赖轻量级的 Canvas 类库 ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表，更是加入了更多丰富的交互功能以及更多的可视化效果，并且对移动端做了深度的优化。

## 9.4. 多维数据支持和丰富视觉编码

除了加入了平行坐标等常见的多维数据可视化工具外，对于传统的散点图等，传入的数据也可以是多个维度的。配合视觉映射组件 Visual Map 提供的丰富的视觉编码，能够将不同维度的数据映射到颜色、大小、透明度、明暗度等不同的视觉通道。

我们可以在图表中加入视觉组件增加用户体验，当鼠标移动到相应位置时，会提示具体的数据，以及数据的权重大小会根据颜色的深浅更加直观的视觉输出。图表的控件可以将图表中的数据通过表格一键呈现，还可以将图表一键导出为图片，便于后续的整理以及邮件等的发送。

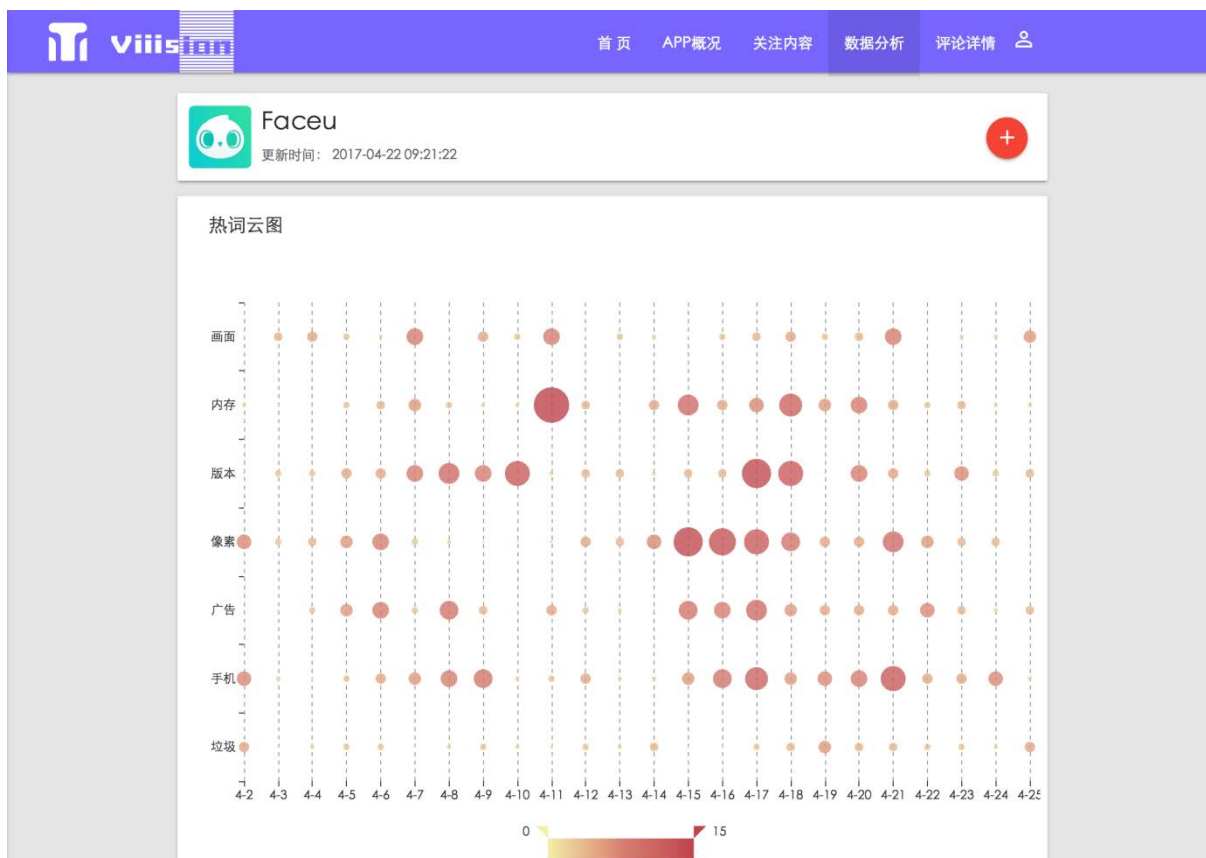


图 9.3 多维信息及视觉特效

## 10. 数据库技术

### 10.1. 可信任的数据库

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件，也是最适配于 PHP 框架 Laravel 的数据库。

MySQL 具有如下特性：

- 1) 使用 C 和 C++ 编写，并使用了多种编译器进行测试，保证源代码的可移植性。
- 2) 支持 AIX、BSDi、FreeBSD、HP-UX、Linux、Mac OS、Novell NetWare、NetBSD、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统。

- 3) 为多种编程语言提供了 API。这些编程语言包括 C、C++、C#、VB.NET、Delphi、Eiffel、Java、Perl、PHP、Python、Ruby 和 Tcl 等。
- 4) 支持多线程，充分利用 CPU 资源，支持多用户。
- 5) 既能够作为一个单独的应用程序在客户端服务器网络环境中运行，也能够作为一个程序库而嵌入到其他的软件中。
- 6) 提供多语言支持，常见的编码如中文的 GB 2312、BIG5，日文的 Shift JIS 等都可以用作数据表名和数据列名。
- 7) 提供 TCP/IP、ODBC 和 JDBC 等多种数据库连接途径。
- 8) 提供用于管理、检查、优化数据库操作的管理工具。
- 9) 可以处理拥有上千万条记录的大型数据库。

## 10.2. 数据库表关系

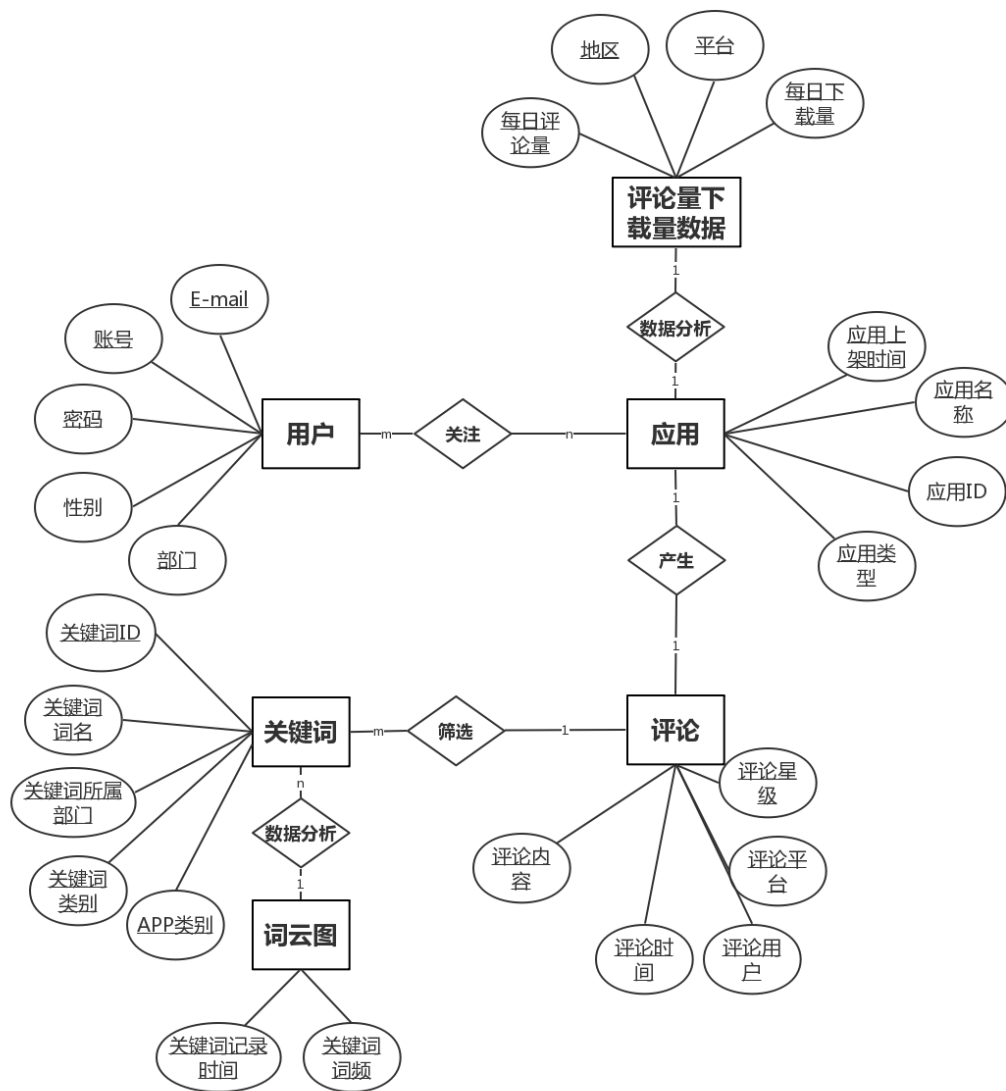


图 10.1 数据库表关系



### 10.3. 数据库表详细设计

表 10.1 app 设计

app		
字段名	数据类型	备注
app_id (PK)	int	应用 ID
app_name	varchar	应用名称
app_category	varchar	应用类型
app_date	date	应用上架日期

表 10.2 app\_info 设计

app_info		
字段名	数据类型	备注
app_info_id (PK)	int	信息记录 ID
app_id (FK)	int	应用 ID
app_platform	varchar	应用所在平台
app_download_count	int	应用下载量
app_comment_count	int	应用评论量
app_district	varchar	数据来源地区
app_date	date	数据对应日期

表 10.3 app\_comment 设计

app_comment		
字段名	数据类型	备注
comment_id (PK)	int	评论 ID
app_id (FK)	int	评论来源应用的 ID
name	varchar	评论用户名
content	varchar	评论内容
score	int	用户打分

<b>date</b>	date	评论日期
<b>platform</b>	varchar	评论来源平台

表 10.4 keywords 设计

keywords		
字段名	数据类型	备注
<b>keyword_id (PK)</b>	int	分词 ID
<b>keyword_name</b>	varchar	关键词名
<b>app_ category</b>	varchar	分词所属 APP 类别
<b>department</b>	varchar	关键词所属部门
<b>keyword_feature</b>	varchar	关键词属性

表 10.5 keywords\_cloud 设计

keywords_cloud		
字段名	数据类型	备注
<b>keyword_info_id (PK)</b>	int	分词信息 ID
<b>keyword_name</b>	varchar	关键词名
<b>count</b>	double	关键词词频
<b>keyword_date</b>	date	关键词所属日期

表 10.6 user 设计

user		
字段名	数据类型	备注
<b>user_id (PK)</b>	int	用户 ID
<b>password</b>	varchar	用户密码
<b>sex</b>	char	用户性别
<b>class</b>	varchar	用户权限
<b>e-mail</b>	varchar	用户邮箱

表 10.7 user\_app 设计

user_app		
字段名	数据类型	备注
user_app_id (PK)	int	用户评论关系 ID
user_id (FK)	int	用户 ID
app_id (FK)	int	应用 ID

## 11. 系统实现

### 11.1. 系统用户端实现

#### 11.1.1. APP 概况

该页面是对相应 APP 的一个总体概述，包括应用描述、基本信息两个方面。采用柱状图和折线图相结合的方式将评论量与下载量在同一图表中呈现，用户可对时间进行选择。



图 11.1 APP 概况

### 11.1.2. 关注内容

该页面根据每个用户的关注内容个性化设计, 用户可以查看自己关注的 APP 类别以及相应的 APP, 点击相应 APP 卡片即可查看该 APP 的相关信息。

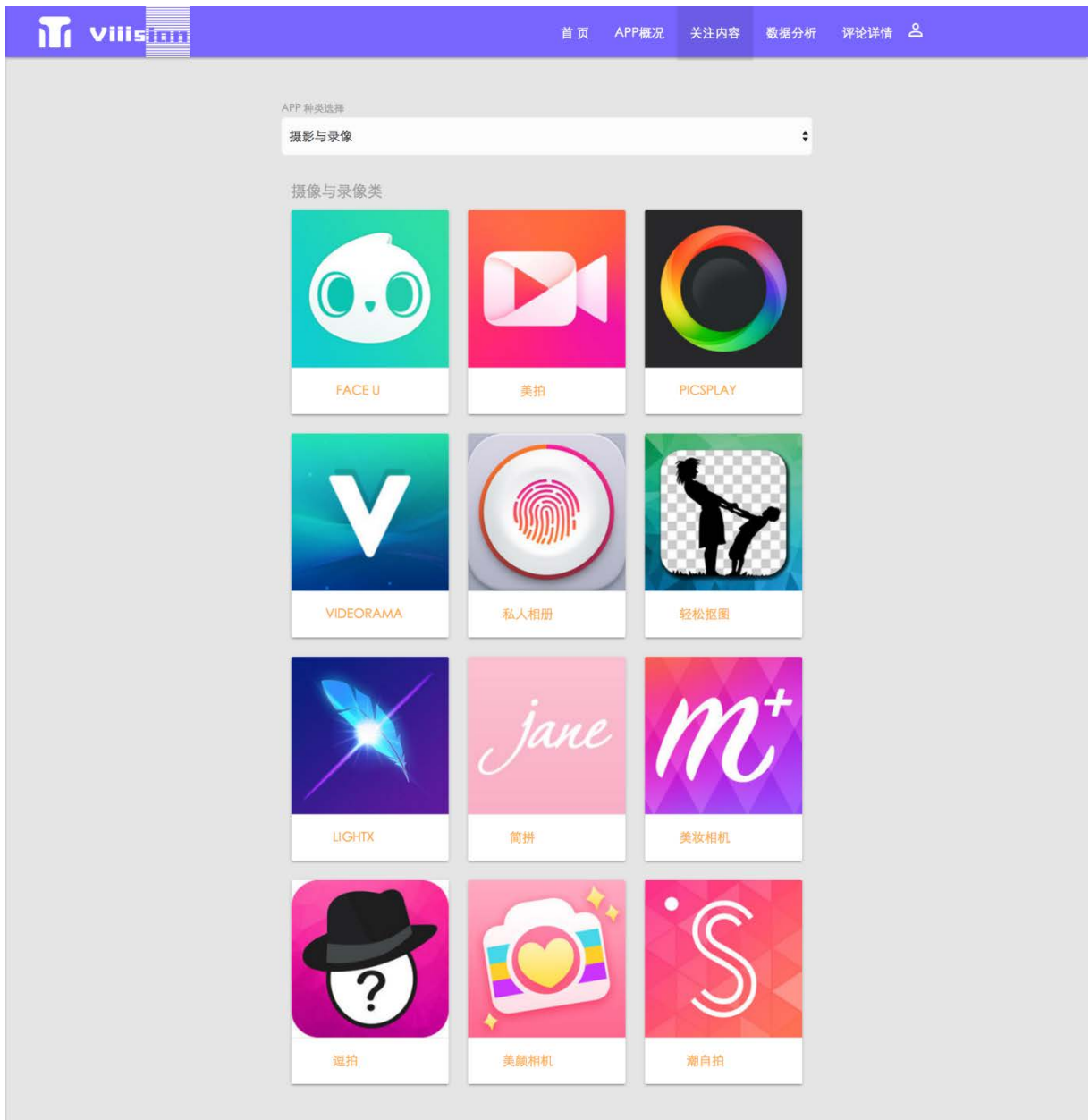


图 11.2 关注内容

### 11.1.3. 数据分析

在该页面中，将统计分析后的结果以多类型的图表形式呈现，下图为词云图，该图集某一时间段内热词排行榜、某一热词词频随时间变化趋势、不同日期的热词分布这三种信息为一体。



图 11.3 词云图

用户可以通过点击相关热词查看包含该热词的好评与差评评论。

#### 热词详情

好评	差评
<p>1.太好了！就是有时候有东西要下，<b>内存</b>占的很大。</p> <p>2.可以 如果<b>内存</b>占小些就OK了</p> <p>3.<b>内存</b>有点大，以后更新的时候，<b>内存</b>小一点 不过还是力推的</p> <p>4.好玩，就是<b>内存</b>大了一点（少给你一颗怕你骄傲@）</p> <p>5.这<b>内存</b>。。好邪门</p> <p>6.刚开始挺喜欢的后来照片就存不到手机上了。不是<b>内存</b>的问题。连续卸载下载了好几次。又好了。不太稳定。。不过还是很喜欢。用了半年了。</p> <p>7.<b>内存</b>比较大，手机比较卡，不过这个软件挺好的，所以给个好评！</p> <p>8.<b>内存</b>大</p> <p>9.还好啦，就是有些占地地方，然后突然有一天贴纸都不见了，还是占着<b>内存</b>所以就删了从下个惩罚就不给五星了？</p> <p>10.特别喜欢，之前下过，但是占<b>内存</b>，所以给了四颗星，不怎么卡</p> <p>11.不太好，<b>内存</b>大</p> <p>12.用了一段时间了，软件很实用，但也有很多缺点，比如闪退、卡，一些特效需要下载这倒不算什么，但特别占<b>内存</b>，然后今天我是因为这个软件更新到一半怎么也不更新了，怎么试都不行，还不能卸载了重下，就是有个正在下载的图标在那里又不能删，特别恶心</p> <p>13.就是有点占<b>内存</b>，耗电快</p> <p>14.好好好好好好，就是<b>内存</b>占的大</p> <p>15.你们每次玩激萌卡的原因是激萌的<b>内存</b>太大了点个赞！</p> <p>16.一直觉得挺好玩，<b>内存</b>也不是很大，平时照相自拍也一直用，闺蜜上次问我这是什么东西真好玩，我就推荐给她，她也很喜欢。最重要的是faceu只要下载上就不用联网了，随时随地自拍美美哒，这是我最喜欢的地方</p>	<p>1 不好，有些都删不了，占<b>内存</b></p> <p>2 以前使用时觉得是拍照神器，可自从更新新版本后，手机就出现闪退问题，卸载后重新下载还是使用的，希望工程师上神们能够帮我解决闪退问题。</p> <p>3 根本就拍不了，只会占<b>内存</b>。</p> <p>4 占<b>内存</b>太大，有点卡，不过功能很好很齐全。??</p> <p>5 <b>内存</b>内存。。。</p> <p>6 根本找不到 我保存下来的视频。不知道咋回事~ 难道是<b>内存</b>不够？</p> <p>7 越来越占<b>内存</b>。不知道开发者怎么想的</p> <p>8 太占<b>内存</b>了！！！！</p> <p>9 卡，但是用别的<b>内存</b>比这还大就是一点不卡</p> <p>10 卡**了不能用啊？，卸载了占<b>内存</b></p> <p>11 真的好卡，之前用不会啊，手机<b>内存</b>还剩2G。就是卡</p> <p>12 一卡一卡的，我手机<b>内存</b>又不是没有，好气</p> <p>13 一般般功能太少<b>内存</b>太大</p> <p>14 挺好的，就是占<b>内存</b></p>

图 11.4 产品特征评论详情

用户可根据区域分布图中的颜色变化可以直观地了解 APP 在全球范围内的分布。



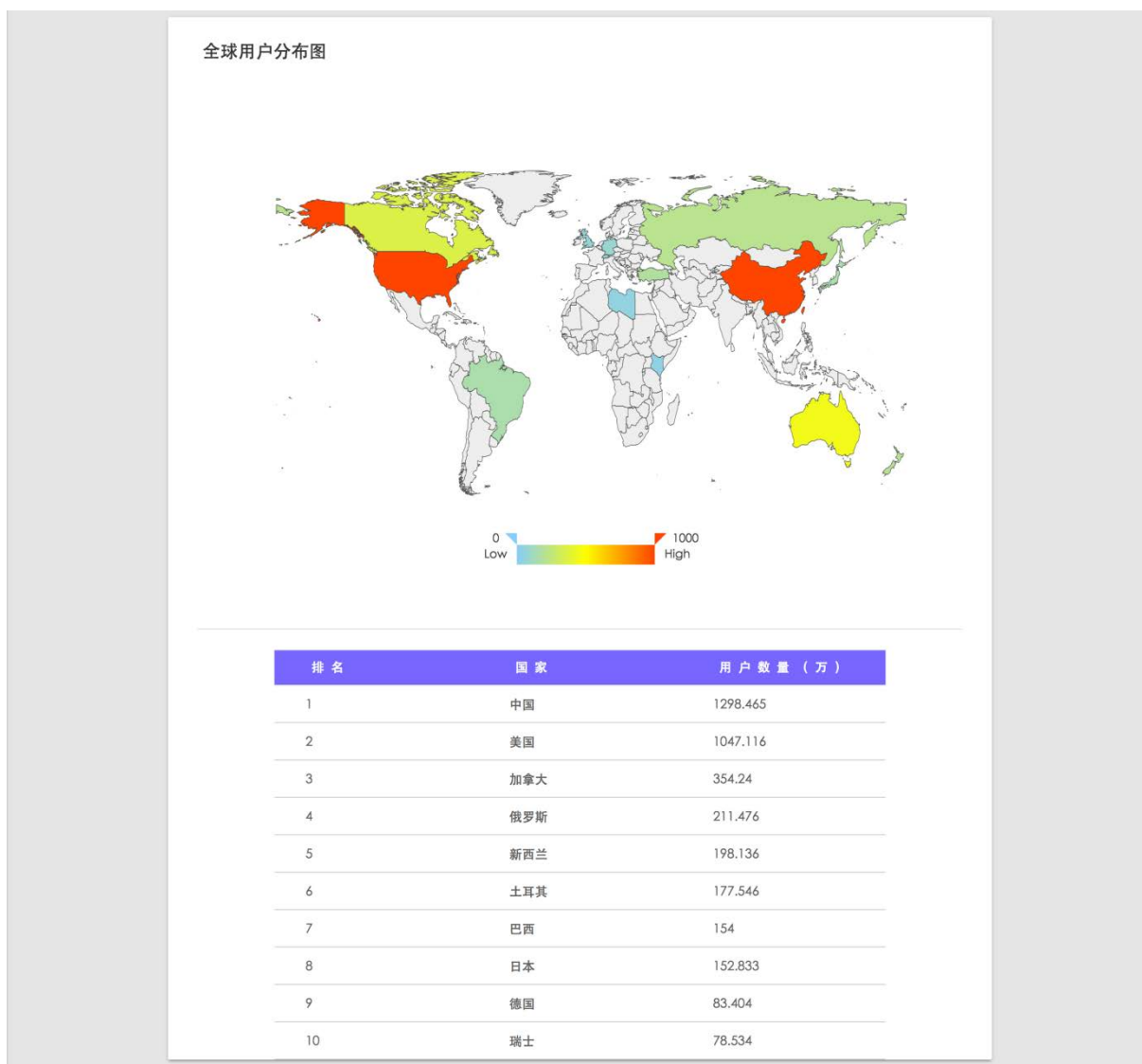


图 11.5 区域分布图

#### 11.1.4. 评论详情

在该页面中，用户可根据时间、平台、星级查看相关评论详情，同时也可根据关键词对评论进行筛选并实高亮显示。

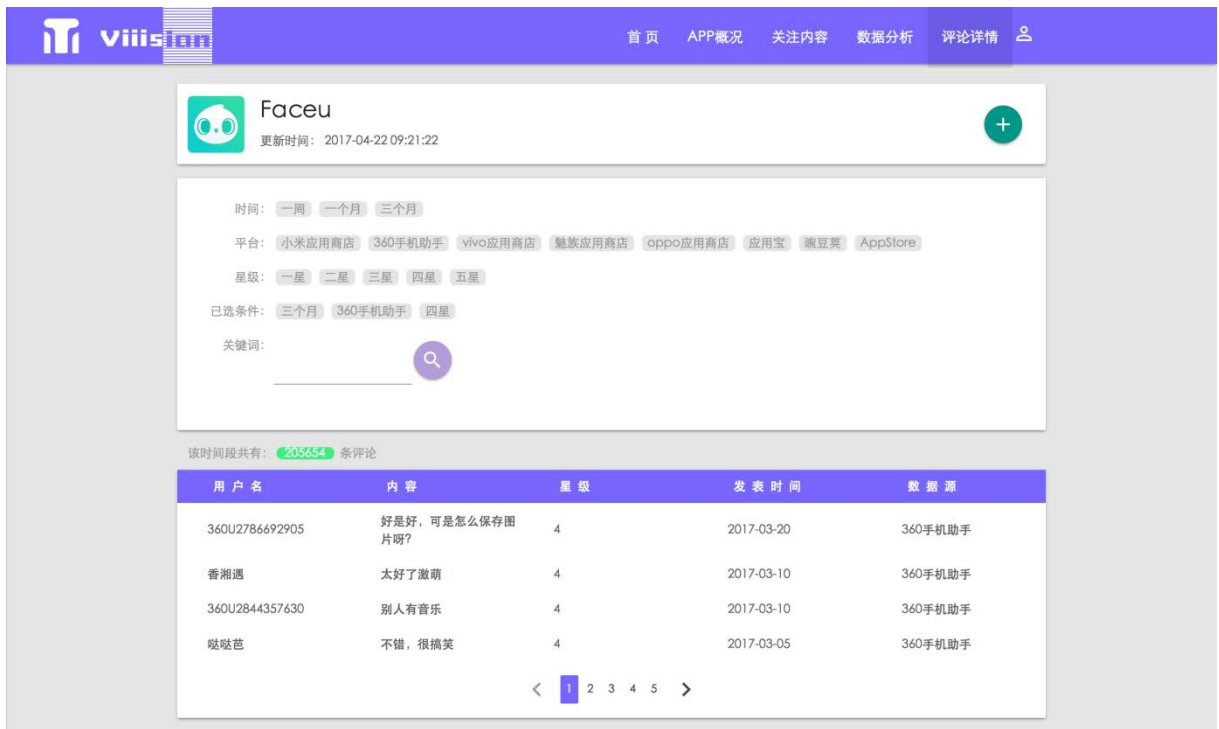


图 11.6 评论筛选图



图 11.7 关键词搜索图

## 11.2. 系统管理员端实现

### 11.2.1. 用户管理

在该页面中，管理员可对用户的基本信息以及其所关注的 APP 进行编辑。

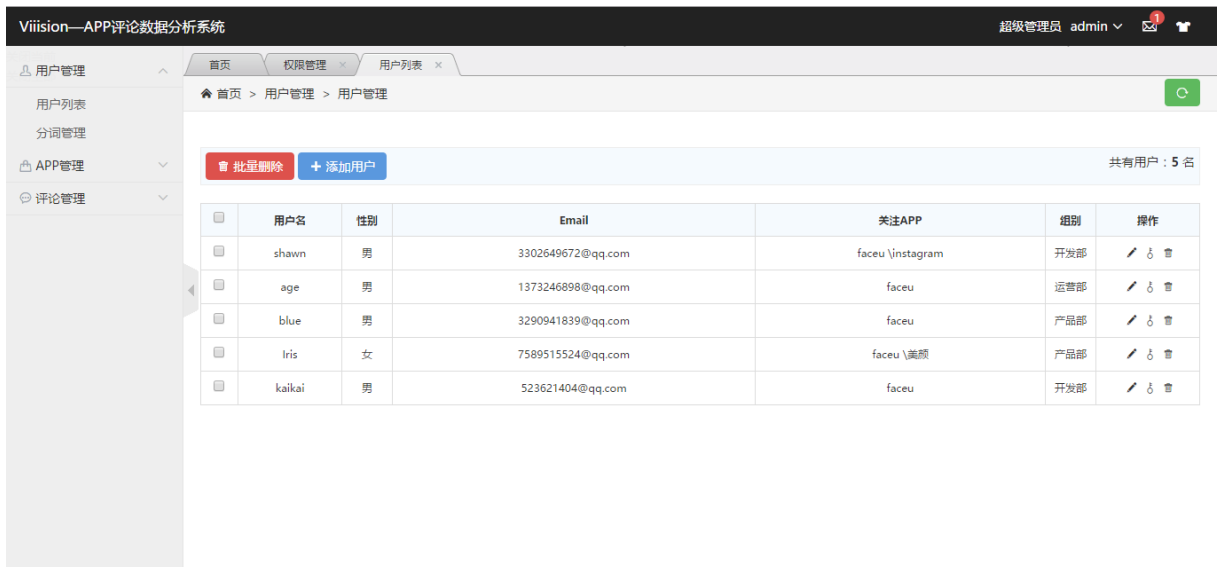


图 11.8 用户管理

### 11.2.2. 分词管理

在该页面中，管理员针对各职能部门的不同需求，分配可查看的分词类别，点击增加分词按钮即可添加分词。

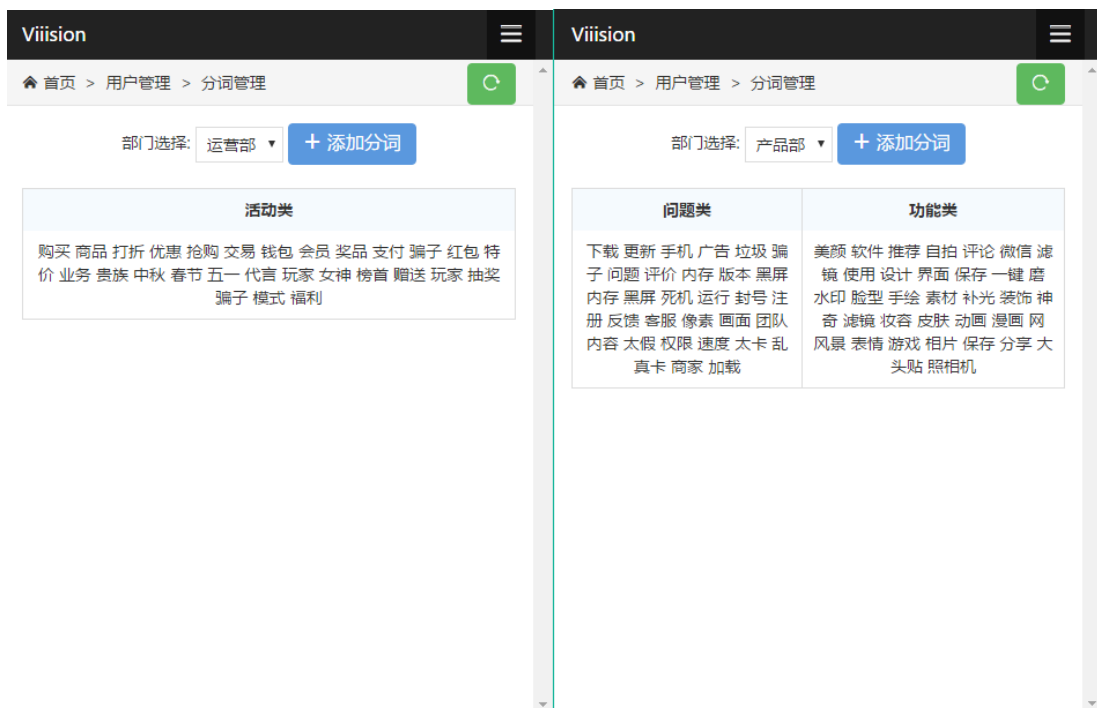


图 11.9 分词管理

### 11.2.3. APP 分类管理

管理员可对 APP 名称及其所属类别进行编辑，并可根据类别进行筛选。

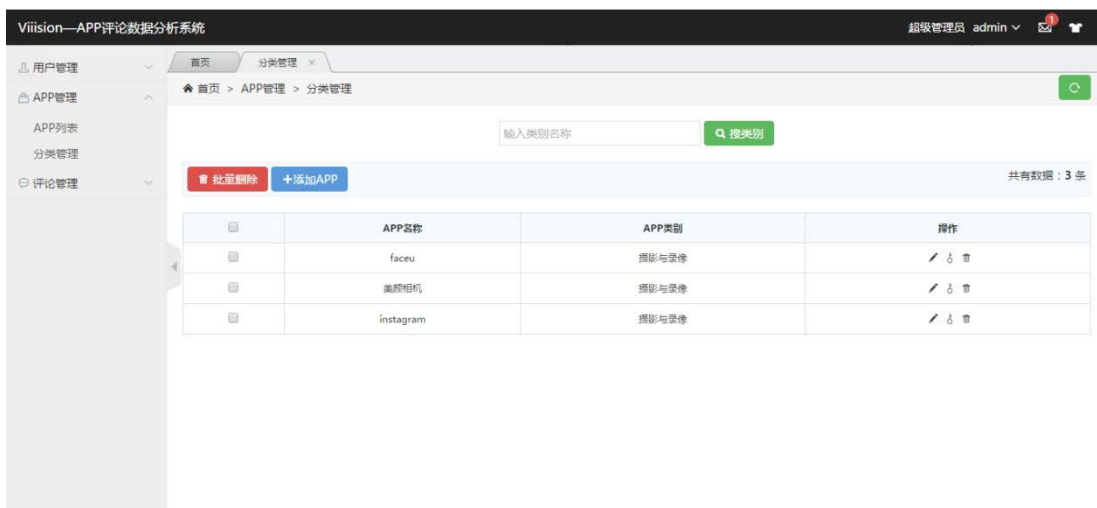


图 11.10 APP 分类管理

#### 11.2.4. APP 列表显示管理

在该页面中，管理员可查看不同 APP 在不同平台下的总体信息。

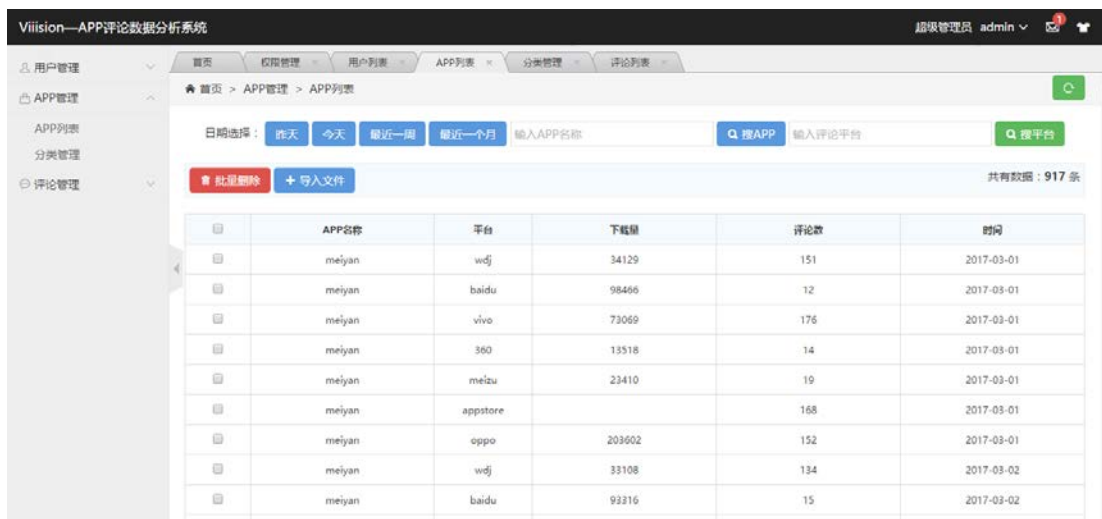


图 11.11 APP 列表

#### 11.2.5. 评论列表管理

管理员通过点击网上导入按钮，输入 APP 名称并配置平台实现网上导入；通过点击本地导入按钮实现本地结构化的数据的导入；通过查看评论详情，可以直接查看导入成功与否。

Viiision—APP评论数据分析系统 超级管理员 admin

用户管理 APP管理 评论管理 评论列表

首页 > 评论管理 > 评论列表

APP选择: meiyan 日期选择: 今天 输入评论平台 搜平台

批量删除 +本地导入 +网上导入 共有数据: 50 条

	内容	平台	评分	时间	操作
<input type="checkbox"/>	好	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	666	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	挺好的，拍照一直用这个	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	很好	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	好	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	太好用了	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	好用，太喜欢了	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	超好用，非常赞同下载。(o>_<o)	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>
<input type="checkbox"/>	好	豌豆荚	0	2017-04-19	<a href="#">/</a> <a href="#">o</a> <a href="#">e</a>

图 11.12 评论详情

Viiision—APP评论数据分析系统 超级管理员 admin

用户管理 APP管理 评论管理 评论列表

平台名称: APP store APP编号: 592331499

平台名称: 华为商城 APP编号: C178302

平台名称: 应用宝 APP编号: com.meitu.meiyancamera

平台名称: 豌豆荚 APP编号: com.meitu.meiyancamera

平台名称: 百度手机助手 APP编号: com.meitu.meiyancamera

平台名称: 360手机助手 APP编号: com.meitu.meiyancamera

平台名称: oppo应用商店 APP编号: 10518558

平台名称: vivo应用商店 APP编号: 48370

平台名称: 魅族应用商店 APP编号: 888353

\*平台名称: \*APP编号:

提交 取消

图 11.13 网上导入