Lyric Search Engine

Team project for course EE208，
基于虾米音乐的歌词+图片搜索引擎

团队分工

- 曹梦奇：组长，scrapy爬虫 + 网页UI优化
- 丁雨晨：网页逻辑搭建，webpy前端基础
- 柳旭东：数据库处理
- 罗英哲：图像哈希检索

环境搭建

- Ubuntu14.04
- scrapy python3.6
- pylucene python2.7 miniconda
- webpy
- PIL for pics
- cv2/opencv

项目简介

- Scrapy爬取虾米音乐2w歌曲的歌词、音乐流派；爬取虾米音乐歌手图片
- 歌词文本搜索，实现对搜索结果进行排序与筛选
- 将歌曲设置ID，根据ID搭建简单的歌曲详细信息网页
- 相似歌曲推荐
- 图像哈希检索，搜索图片库中歌手图片，返回该歌手的歌曲

项目历史与项目前景

现在互联网上歌词搜索引擎存在版权归属、UI设计、运维等各种问题，而且各大音乐平台的资源都是闭环的，不能实现资源的共享。接下来我们打算对QQ音乐，咪咕音乐，网易云音乐。本项目仅用作课程作业，绝不用于商业。

技术细节

曹梦奇

> - Scrapy
> 爬取歌曲时使用Scrapy创建两个Spider，第一个负责爬取歌曲url结果导入csv文件，第二个则是读取csv文件的所有url，对每一首歌分别利用xpath进行信息抓取。爬取歌手图片也是获得歌手url，然后对每一个歌手页面抓取对应位置的pic。
> 由于虾米音乐反爬虫政策，我在middlewares.py中自定义函数process_request( )对爬虫进行时间复杂度上的优化；在setting.py中设置代理IP。虾米音乐提供的信息大部分为static，所以基本不需要抓包。
> - UI优化
> 外部CSS文件中对网页排版进行设计，利用JavaScript实现主页上功能介绍时的滑动效果、光标移动到链接上时颜色改变等效果。

罗英哲

> - 先加载xml文件生成级联分类器face_cascade，然后用这个级联分类器对灰度图进行检测，返回值即图片中所有人脸的坐标（左上角，右下角坐标），将图片得出的人脸小图存成图片文件。将所有爬到的图片根据此方法转换成人脸小图并存储起来。
> - 根据HASH算法，在最后一次实验作业的基础上进行改进，根据BGR三种颜色分布得到3*4*4 = 48维的图片特征向量，再将该特征向量进行简化，然后建立HASH字典，并调用pickle将字典以及两个向量均存入pkl文件。
> - 根据要进行搜索的图片，切出人脸图，得到特征向量以及简化后的特征向量，然后在存在pkl文件中的HASH字典中进行查找，找到所属类别后再进行特征向量的比对，由于每次生成的特征向量会有一定区别，因此在比对时加入了误差忽略处理，增强了鲁棒性，最后对于大部分歌手图片都可以完成快速正确的搜索。
> - 由于对于图片相似度的计算时间过于长等原因，没有实现对给定图片库之外的歌手图片进行识别，这一部分还有待改进。

柳旭东

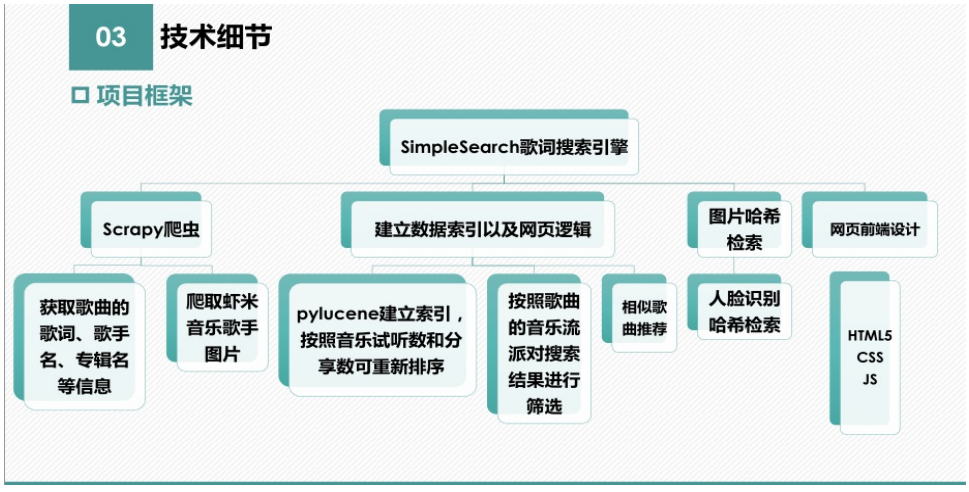
> - 索引建立
> 将爬虫数据存储在csv中，每首歌在索引中的信息包括id号、专辑名、歌手名、歌曲名、歌曲在虾米音乐网站对应的url、歌曲语言、试听数、收藏数、分享数、专辑图片的地址、歌曲的流派以及歌词。使用lucene建立索引，对于中文的歌词，需要先使用jieba分词工具分词后再存入索引。
> - 搜索
> 做网页的同学考虑到如果要将符合要求的歌曲的所有需要显示的信息都在文本搜索后一起传给显示歌曲详细信息的网页时会因为庞大的数据量大大地影响效率，所以我们采用了二次搜索的方法：在第一次基于歌词搜索后得到符合要求的歌曲的简单信息（包括id号），显示在主结果界面。在用户点击每一首歌的链接时，通过这首歌的id在索引中进行检索，得到这首歌的相关信息，然后在歌曲详细信息的网页上显示。每首歌曲的id号也是以类似文本的形式保存在索引中，并保证相同的长度（根据数据库中总的歌曲数确定）。

> - 关于相似歌曲
>   基于每一首歌，遍历数据库中其他所有的歌曲。歌手相同，相同标签数多的两首歌曲则被认为是相似度高，基于此原则在遍历过程中对其他所有歌曲打分。对于每首歌，将与其最相似的三首歌（得分最高的三首歌）的id号（也会被用于后续搜索该歌曲）和歌曲名（即这首歌的相似歌曲信息）存入索引。
> - 关于按热度排序与按标签筛选
>   在主搜索结果界面显示的歌曲以s及歌曲的相关信息都存放在一个列表（该列表始终保存当前显示的歌曲信息）中，按热度排序只需基于每首歌的试听数/分享数，将列表中的歌曲及相关信息重新排序，再将刷新后的列表重新传给刷新后的网页显示。按标签筛选时，先统计出所显示的歌曲的全部标签，用户点击相应的标签按钮后，只保留带有该标签的歌曲及相关信息在列表中，再将刷新后的列表传给网页。

丁雨晨

> 利用w ebpy实现了预期的搜索引擎目标--对用户的搜索请求进行反应，返回正确结果；优化了网页数据传递结构，加快搜索速度；整合资源，提供对后端的接口，简化开发流程，提供开发效率。

成果展示（答辩PPT）



04 成果展示

文本搜索结果页面

能实现按照当前音乐收听量或分享数对已有结果进行重新排序检索；

能实现按照音乐流派对搜索结果进行筛选；

歌词显示页面

显示歌曲的曲名、歌手名、专辑名、专辑图片、全部歌词、播放链接、右侧提供三首推荐的相似歌曲

图片搜索结果页面

识别上传图片中对应哪位歌手，以该歌手名作为输入进行歌曲搜索