大作业实验报告

成员：聂鹏博、林浩东、裘索、席奇

大作业主题：根据图片或经典台词搜索电影

一、实验分工：

1、聂鹏博（学号：516030910482）：网络爬虫的实现，数据的搜集。

2、林浩东（学号：516030910480）：根据得到的数据建立搜索系统。

3、裘索（学号：516030910484）：搜索引擎界面的 html 文件的实现。

4、席奇（学号：516030910487）：联系搜索引擎界面与搜索系统的 code.py 的实现。

二、实验内容：

1、网络爬虫：

对应文件夹：‘crawler’

1.1 图片爬虫：

（1）代码文件为‘imgcrawler.py’，爬取的是 6v 电影网上的台词，采用的基本方法是传统爬虫的方法，在爬到每个网站时首先判断是否是所要找的电影网页，若是，则将该网站上的图片保存下来。

（2）注意事项：首先是要防止爬到别的网站去，因此爬的时候必须要爬以 www.6vhao.tv 开头的网站。其次是要防反爬虫，除了将 headers 填充完整外，还需要在每次爬取时停顿随机时间（2~4s）。

（3）这次爬取的网页有一万多个，图片有三万到四万张。

（4）爬虫生成的文件为，‘index.txt’：保存 url、电影名和相应的文件夹名称。文件夹‘html‘，其中每个文件夹代表一部电影，里面是该电影的图片和一个关于其中图片的’index.txt’(该 index 中是图片名字和图片的 url)

1.2 台词爬虫：

（1）代码文件为‘txtcrawler.py’，台词爬虫爬取的是句子迷网站，采用的是新的方法：先爬取电影的索引页，然后根据其上的电影台词链接进入相关页面，一页一页的将电影的经典台词保存，然后再返回索引页，爬取下一部电影。等到一个电影的索引页全部爬完之后，再进入下一个索引页，直到全部爬完。

（2）由于网站的反爬虫功能比较强大，这里采用了使用代理 ip 的方法进行爬虫，ip 的生成和检测其可用性的代码文件分别为‘ip.py’和‘iptest.py’。

（3）爬虫生成的文件为，‘index.txt’：保存 url、电影名和相应的文件名。文件夹‘html’：其中每一个文件储存的是每一部电影的经典台词。

（4）由于句子迷网站本身的电影的经典台词就比较少，只有 100 部，因此爬到的数据也就不多。

2、搜索系统与 code.py：

对应文件夹：‘index_building’与‘core_code’

文字搜索：通过 lucene 建立爬虫数据的索引文件，再根据待搜索内容进行检索。利用学习到的 lucene 知识完成。其中应用结巴函数来进行分词。与之前学习内容基本类似。

对应‘txt_index’文件夹，其中 Indexfile.py 用于建立文字索引，SearchFile.py 用于搜索，用在 web 中做结果处理。

图片检索：对于图片的匹配，采用 hsv 进行匹配，即考虑色相、饱和度、明度的综

合匹配。并且对于直方图的统计是进行整张图的匹配，但是对于图片中央和边缘的颜色分布通常是不一致的所以将图像分成不同的区域进行统计。基于这样的颜色算符建立图片描述符的索引文件，在通过对应的查询代码进行搜索。

对应'img_index'文件夹，其中 Index.py 用于建立图片索引，search.py 用于查询，用在 web 中做结果处理。

界面设计代码实现:将搜索引擎的查询功能当做函数对从 html 处取得的数据进行处理，将结果返回 html 文件。

code.py 用于界面代码处理。

3、 页面设计：

我们的搜索引擎使用了 4 个 html 文件对应四个界面，正如命名的那样分别是图片搜索，句子搜索，图片搜索结果和句子搜索结果。页面设计使用了 div+css 的结构，以类选择器的设计为主，结合了内部样式表进行界面的呈现。、

对于搜索界面，使用了超链接的方式来做到两种不同类型搜索的切换，图片搜索的情况我们需要用户输入的是该图在本机上所处的路径，输入框是用 form 完成的。在类选择器中，设定了各种关于界面主体，标题，文字，输入框的属性，使它们达到我们想要的显示效果，并用绝对位置进行定位，确保不会出错，在多次尝试后将各个元素摆到它们应处的位置上。并根据 css 使用手册对字体的样式，大小，粗细，颜色做出种种优化。

对于搜索结果界面，由于考虑到用户很有可能并不只做一次搜索，于是我们在结果界面同样设置了输入框以及供切换的超链接以方便用户进行进一步搜索。结果界面使用 div 对页面分块并设置了背景框，便于用户分辨不同结果，页面的结果包括本地的该图片以及搜索到的图片对应的电影名以及相应的海报或剧照，电影名包含了前往所在电影的电影网站链接；句子的结果包含了电影名，电影中匹配到的句子，以及指向该电影句子迷网站的链接，用户可以通过点击该电影名直接前往该网站。

做界面时很伤脑筋的一点就是 web.py 无法上传本地的背景图片，使用 background-image 不能起作用，致使所做的页面缺乏背景，整个界面都因此变得不美观。后来想到了解决办法，就是先将我们要使用的本地背景图片上传到网上，在 body 中再用 src 打开网上对应的图片网址，将该图片上下左右充满，再将其置于整个页面的最底层不干扰其它的标题文字，这样我们便成功地变相做到了插入本地的背景图片。

这样一来，界面的工作完成了，剩下的就是做好界面与主程序的接口了。

三、程序运行与结果演示：

1、

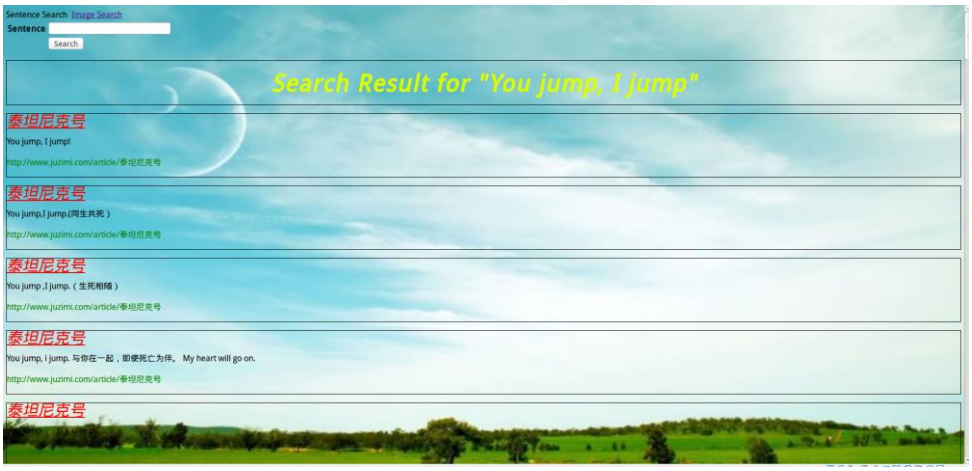
对应文件夹：'core_code'

运行时只需在 ubuntu 终端运行 code.py 即可。

打开的网页界面如下：

2、台词搜索电影：

选择 Sentence Search，并输入想要搜索的台词，即可进行搜索，结果如下：



3、图片搜索电影：

在搜索前，需将所要搜索的图片放入‘core_code’文件夹中的‘target’文件夹中，然后选择 Image Search，并在搜索框中输入图片的名字即可进行搜索（由于图片数量较多，运行时间

可能较长，大约一至二分钟)。结果如下：