**外卖搜索引擎**

网页爬虫

这次的爬虫实验我主要是爬取美团、百度和饿了么的网站来获取相关的信息。首先是爬取的过程中遇到了一些困难。爬取美团网站、饿了么网站容易出现403的错误，IP被封住了，尽管后来尝试了模拟浏览器信息登录，降低访问频率的方式，仍然经常出错。而爬取百度的网站却遇到了部分想获得的信息会被：：before之类的伪元素封存。多次尝试后，我换了一种新的思路。首先我先用一个小程序将这些网页的所有内容都以str的形式保存下来，而非直接urlopen，这样就可以绕过很多困难来解决问题，并且将一些动态容易出错的问题变成定态的内容，使程序更稳定，由于str的形式，所以伪元素的问题也就不存在了。然后我做了两个爬虫程序主要是是爬取闵行和徐汇交大附近的外卖商户和一些种类的食品信息，由于网页都是比较规范的，所以正则表达式是可以非常方便的找到相关信息，只要改变一些参数，就能够对多网页进行同样的操作，将得到的相关信息做成一个文本，为下面做索引做铺垫。

在爬虫实验中，我主要的感受是面对复杂的网络环境和网络反爬虫设置，一定不要拘泥于一种方式，可以找其他方式来找到出路，获得我们想要的信息，我们的最终目的是获得网页中的某些信息，而不是为了爬虫而爬虫。

算法实现：首先，读入主程序外小程序读出的str形式的网页内容；然后，用beautifulsoup将网页内容变成能够用正则表达式读取的形式；接着，在相关网页上查看我们想要的信息藏在那些标签下，将其读出来，并进行一定的转换来得到我们想要的形式，储存在文本格式下，并将一些相关网站的网页内容读出来放在html下，以备之后在做index，后经过改进，可以不用做html文件，直接将所有想要获取的信息读出来，这个改进可以节省大量时间。最后，将所有内容分成不同的类做成多个result.txt文件

索引建立与图像搜索

　　索引建立过程主要是把爬到的数据进行处理和保存。相对而言，每一个店铺每一个商品对应的数据是比较多的，我以店铺和商品的名称作为可以搜索的数据，店铺中的数据还包括店铺网址，店铺评分，月售量，起售价，配送费，预计时间和图片地址，商品的数据还包括商品名称，商品对应的网址，商品所在的店家名称，价格，推荐人数和图片地址，通过对于这些索引数据的搜索和使用，可以建立起一个较为完整和丰富的搜索结果页面。

　　由于在数据爬取过程中存在的一些问题，可能是网页本身存在的问题，也可能是数据爬取手段的限制，导致获取的数据中存在一些空缺。以商家的数据为例，一些店铺的预计时间数据缺失，确认后发现这些店确实没有这个数据，属于未知。所以我在建立索引过程中也对这些缺失的数据进行了补全。

　　为了页面设计的同学调用索引中的数据，尤其是方便排序，我需要在建立索引时把一部分字符串形式的数据去掉文字，简化为纯数字。例如：店家的月售量，起售，配送费，商品的价格，推荐人数等。

　　还遇到的问题是由于一部分店家和商品的名字里带有空格，导致难以锁定在一行上的位置。对此，我采取的办法是对每一行从后往前读取数据，由于含空格的数据都是在每一行的最前面，可以借此确定。

　　图像匹配的实现：我调用了Pillow库中的image\_match()函数来计算两张图的相似程度。我们预先在常见的食物种类里添加了十张图片，通过函数计算出每一张图与待搜索图像的相似度，并输出匹配度最高的图，即函数返回值最高的图所在的类别的编号，并在结果网页中按一定顺序显示食品。

由于一些食物的外形实在过于相似，而同类的食品也存在不同的外形和颜色，我尝试和学习了各种图片相似度计算的方法，但都不够准确，存在误差，希望+未来能够改进。

网页界面和网页搜索

本次实验我负责的是商品商家搜索与网页界面设计的部分。在搜索部分，主要使用了lucene语料库以及排序算法。网页设计则是由html+css+webpy制作的。

借助小组成员制作的索引做接口，可以简单地调用爬到的网页信息。然后将信息用lucene内置的analyzer和seacher，找到name项与关键字匹配的数据，并存入结果。在测试中，发现偶尔会有歧义的匹配结果，例如“面包”会匹配到面类等，为此，我在调用函数时对输入的关键字做了一点加工。

由于之前对网页制作不太了解，我去观察了百度，美团和饿了么三个目标网站的页面html信息，发现对于外卖的搜索结果，传统搜索引擎的信息排列方式无法直观快速地让用户获取有效信息，因此我也决定使用卡式布局来呈现搜索结果。实际制作时，遇到了几个问题，其中大多是由于对html的理解错误导致的，例如一开始遇到的资源调用问题，打开的网页调用不到设置的图片，是因为没有把图片等资源放进static文件夹下，自然也无法作为静态资源调用。与之类似的还有表单，样式表排版等问题，在网上查找了css和webpy的教程（www.w3school.com.cn与webpy.org）后，基本都得到了解决。html的使用的特点是，关于格式，绘图，版式等信息都存储在css样式表中，html程序只负责元素的创建和摆放，但通过表格元素，也能实现对元素位置的控制。另外，之前我曾经以为网页之间的跳转是通过<a href=…></a>来链接到目标所在的地址来完成的，后来发现事实上，这个过程是通过渲染(render)与网页模板相同的页面来实现的。

在创建搜索框元素时，我尝试使用了一些css的绘图方法，制作了一个带有动画效果的搜索框，并将这一效果应用到了标签按钮上。在排列搜索结果卡片时，不时会出现一排空白的情况，经检查，是由于卡片布局内字符串行数不同导致的，因此限制了字符串长度的最大值。另外，在制作盒子模型时，margin, padding, position三者的区别在于，margin主要是指该元素与相邻元素之间的距离，padding主要是指元素内容与块级元素设置边框之间的距离，这里不包含边框的宽度，而position主要分relative/absolute，是根据父级元素设置相应的位置的变量。

在运行过程中，曾经出现过点击“后退”后无法再次提交表单的问题，而且按照ppt上的方法，添加vm\_env=lucene.initVM()也没有用。多次尝试后，发现将初始化单独写在一个文件里能解决这个问题，原因尚不明。

虽然还有很多没能做到完美的地方，但这毕竟是我第一次尝试制作搜索网页，让人感觉虽然有瑕疵，但也有做的比较好的地方，最重要的是，我初步理解了制作网页的整体流程，下次不会再花费大量时间去应对由于理解错误而导致的低级问题了。现在我可以完成网页的跳转，排版，简单绘图，按钮点击/鼠标移过效果，提交表单并用程序处理信息等功能，如果之后还有机会接触网页html的编写，应该可以掌握的更加熟练。

小组分工：

网络爬虫：赵雪毅

索引建立与图像搜索：张煜辉

网页界面和网页搜索：张超然