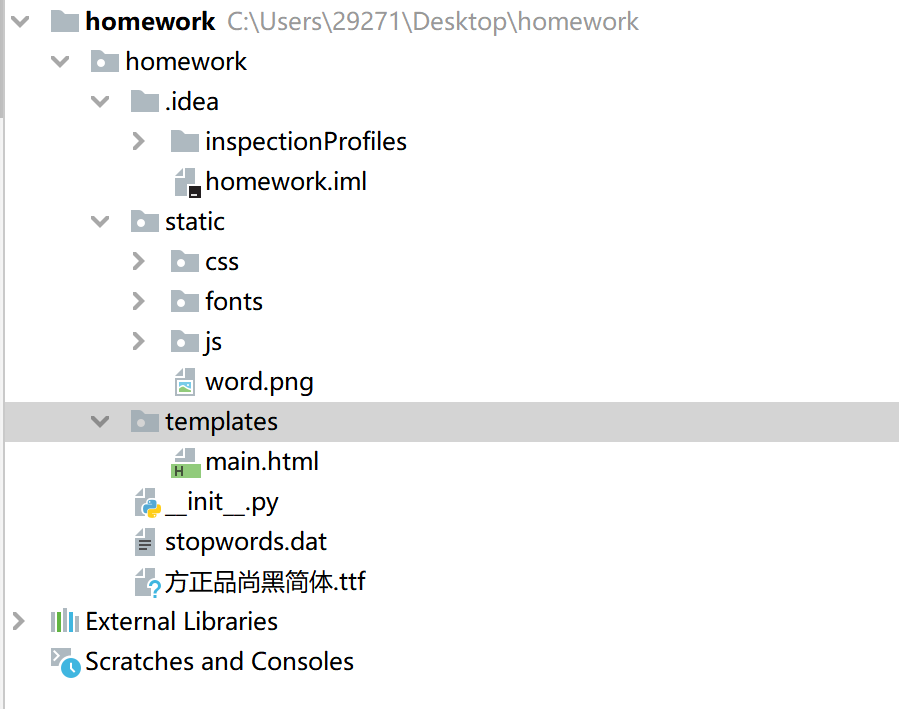
1.实验前准备：

1. 安装flask框架：

使用命令：pip install flask

1. 安装bootstrap：登录bootstrap官网下载bootstrap-3.3.7-dist压缩包
2. 下载stanfordcorenlp, jiagu, matplotlib, wordcloud等工具包。
3. 下载停用词表
4. 项目工程结构及说明：



1. \_init\_.py: 存放所有的python文本处理函数 ，包括实现“分词”、“词性标注”、“命名实体识别”、“情感分类”和“词云展示”的函数，也是函数运行的入口
2. templates文件夹下：main.html存放前端代码和接口
3. static文件夹下：存放前端样式（来自bootstrap的压缩包中的css, fonts, js包
4. Stopwords.dat：停用词表
5. 前端部分：
6. Bootstrap的使用:

Bootstrap是前端开发中比较受欢迎的框架，简洁且灵活。它基于HTML、CSS和JavaScript，HTML定义页面元素，CSS定义页面布局，而JavaScript负责页面元素的响应。Bootstrap将HTML、CSS和JavaScript封装成一个个功能组件，用起来简单，粗暴。

本项目使用了Bootstrap导航栏、标签等元素。其中，导航栏的作用是切换不同的文字处理功能，点击导航栏不同的按钮，可以跳转到相应的功能。标签用于提示输入区和输出区

Bootstrap 中的导航组件都依赖同一个 .nav 类，状态类也是共用的。改变修饰类可以改变样式。在标签页上使用导航需要依赖 JavaScript 标签页插件。由于标签页需要控制内容区的展示，因此，必须使用标签页组件的 JavaScript 插件。另外还要添加 role 和 ARIA 属性。第二步是确保导航组件的可访问性，在使用导航组件实现导航条功能时，在 <ul> 的最外侧的逻辑父元素上添加 role="navigation" 属性。

1. 功能展示区布局：布局分为left类和right类。left类位于左侧包含输入提示信息及输入框，右侧为结果展示。

首页和导航栏展示：



功能区展示：

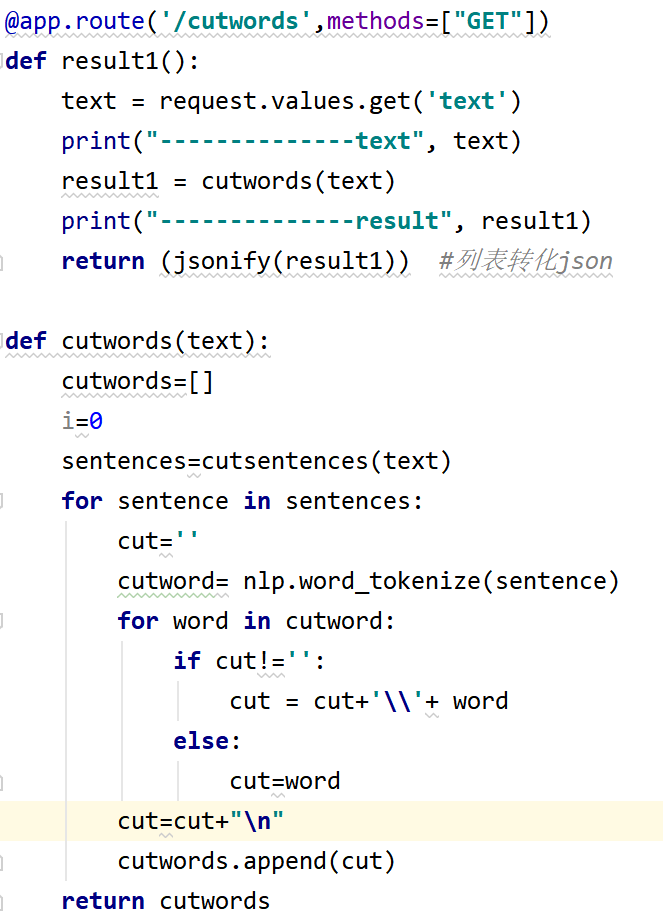


1. 后端部分：

本项目一共实现“分词”、“词性标注”、“命名实体识别”、“情感分类”和“词云展示”五个功能。以下是对实现相应功能的模块的分别说明：

1. **分词**：分词指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。

分词部分代码如图：



当用户点击按钮时，按钮类为Class=”selectButton1”, 从而跳转到url=http://127.0.0.1:5000/cutwords, 触发cutwords函数。

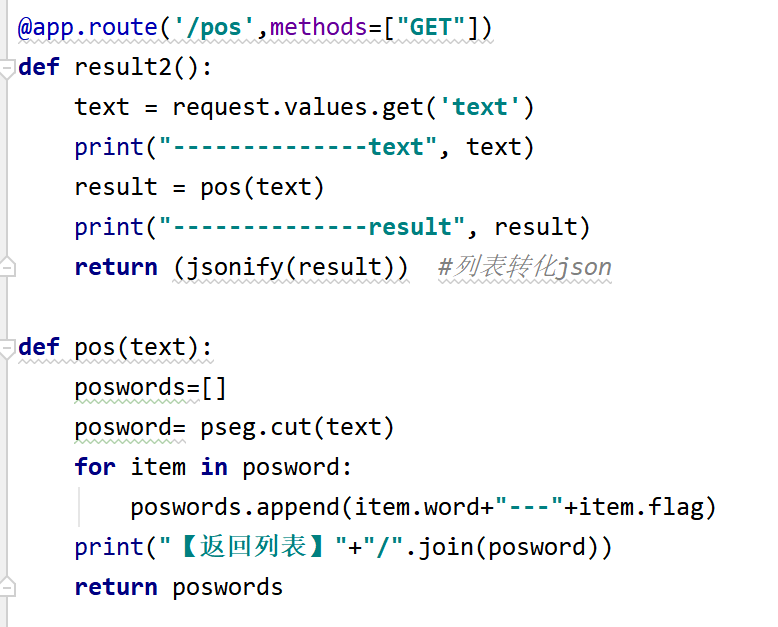
在cutwords函数中，将前端传入的文本放入sentences变量中，使用nlp包中word\_tokenize(sentence)去除文本里所有的空格后获取分词。在获取的每个分词后面自动添加回车实现换行，返回分词组到result1()函数，之后result1函数将列表转化为json形式，返回给前端展示。

前端处理结果如图所示：



1. **词性标注**： 一个对句中的每个标识符分配词类（如名词、动词、形容词等）标记的过程。

词性标注部分代码如图：



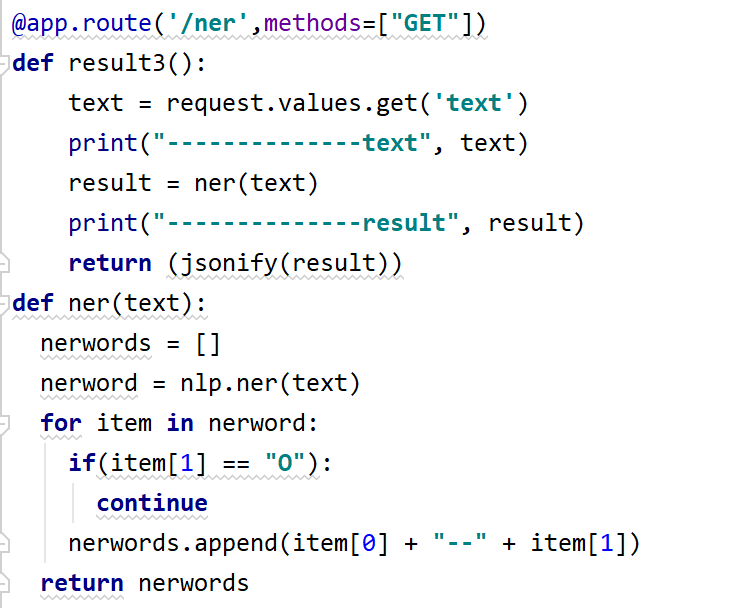
当用户点击按钮时，按钮类为Class=”selectButton2”, 从而跳转到url=http://127.0.0.1:5000/pos,触发pos函数。在pos函数中，使用jieba分词pseg.cut(text)对文本进行分词，使用for循环在获取的每个分词后面添加词性标注(item.flag)，返回分词组到result2()函数，之后result2函数将列表转化为json形式，返回给前端展示。

前端处理结果如图所示：



1. **命名实体识别**：命名实体识别是NLP里的一项很基础的任务，就是指从文本中识别出命名性指称项，为关系抽取等任务做铺垫。狭义上，是识别出人命、地名和组织机构名这三类命名实体。

命名体识别部分代码如图所示：



当用户点击按钮时，按钮类为Class=”selectButton3”, 从而跳转到url=http://127.0.0.1:5000/ner,这里使用stanford包中的ner函数进行处理。在ner函数中，使用stanford的nlp包，调用ner函数对从前端传入的文本进行命名体识别，在获取的每个词后面添加命名体识别结果，返回分词组到result3()函数，之后result3函数将列表转化为json形式，返回给前端展示。

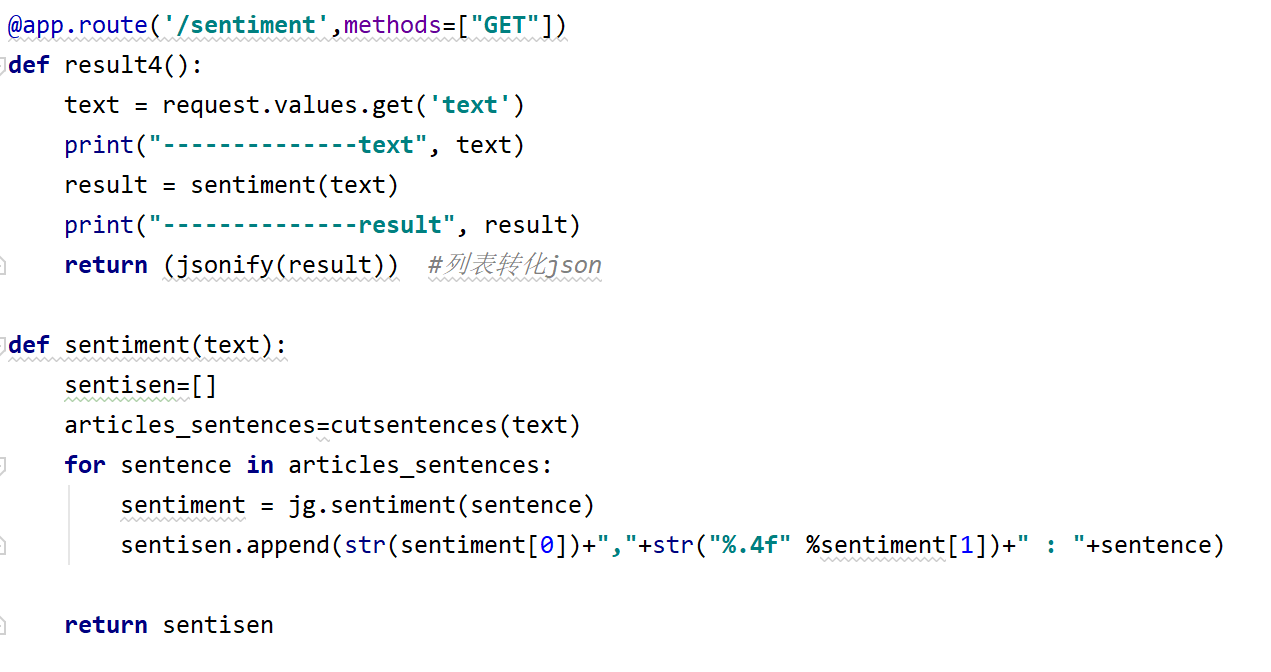
由于命名体识别的目的是识别出有效、重要的命名体，而文本处理结果中大量的”O”命名体没有特别大的意义。为了更有效地展示有用信息，我对命名体识别结果进行了处理，即去除了命名体识别为O的结果。

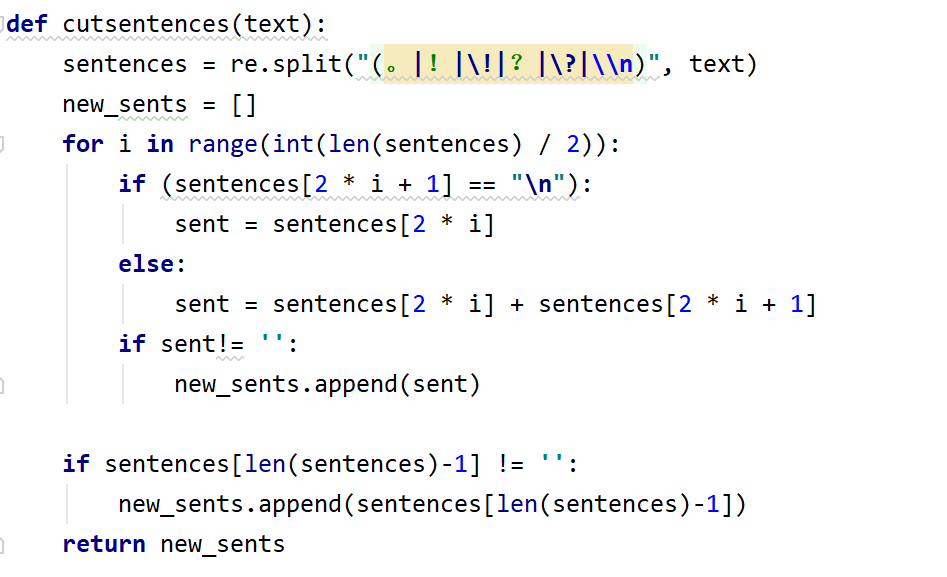
前端处理结果如图所示：

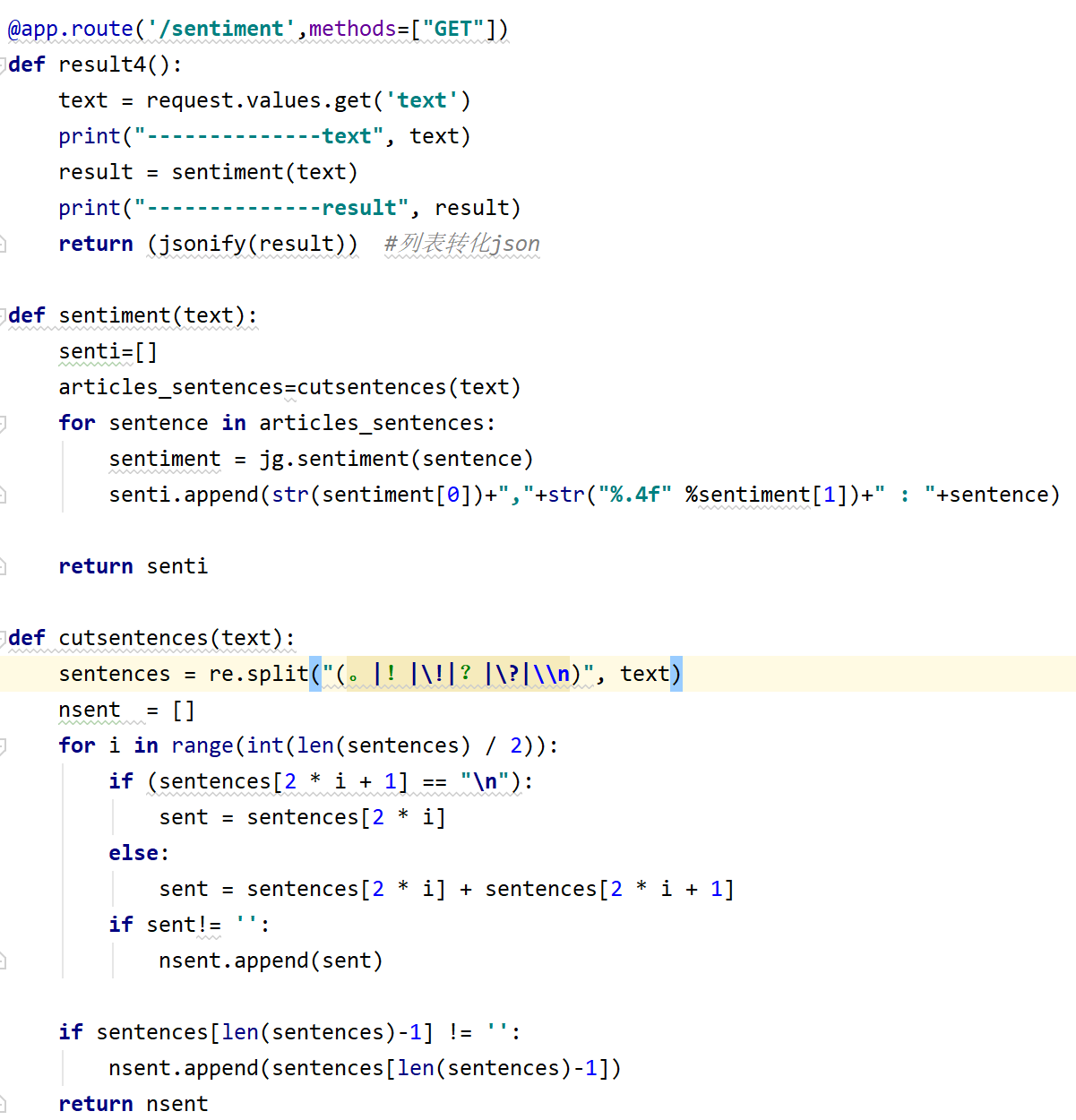


1. **情感分类**：情感分类是指根据文本所表达的含义和情感信息将文本划分成褒扬的或贬义的两种或几种类型，是对文本作者倾向性和观点、态度的划分，因此有时也称倾向性分析。

情感分类部分代码如图所示：







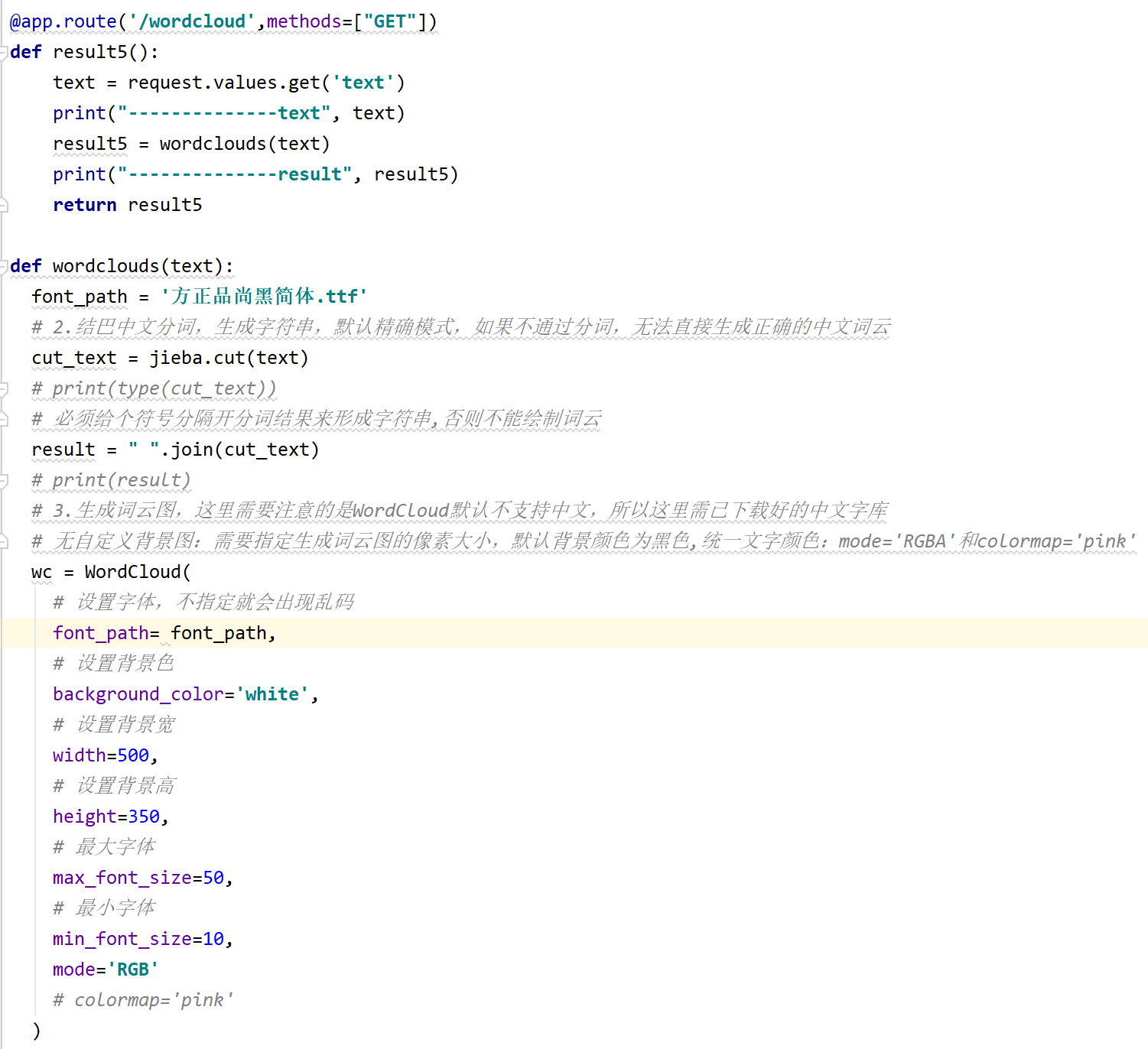
re.split()按照匹配的字符串将前端传来的文本进行分割。之后进行每个词的判断。词性类别如果是积极词语，则检测前后词语，如果前一词为程度副词，则+2，前一词为否定词、消极词或后一词为消极词，则-1，其他情况+1。如果是消极词语，则检测前一词语，如果前一词为程度副词，则+2，前一词为否定词则+1，其他情况-1。最后输出权值。

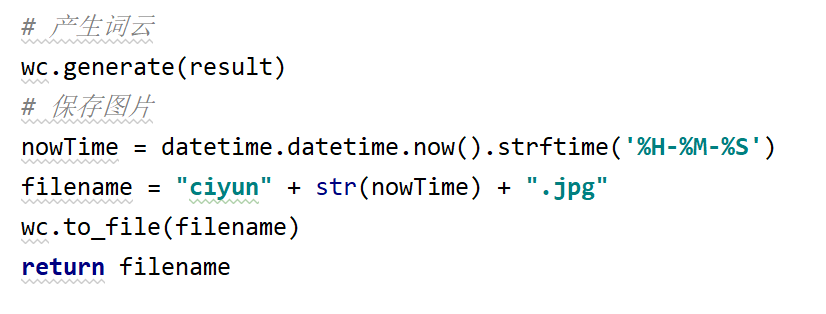
前端处理结果如图所示：



1. **词云展示**：

词云部分代码如图所示





代码思路如下：

1、读入前端传入的txt文本数据。

2、使用结巴中文分词，生成字符串默认精确模式，如果不通过分词，无法直接生成正确的中文词云。

3、无自定义背景图：需要指定生成词云的图的像素大小，设置字体以防止乱码。

4、生成词云图并保存至本地：由于WordCloud默认不支持中文，所以这里需已下载好的中文字库。

4、将生成的文件路径返回前端。前端通过语句：<img src=../" + data + " width='400'/>

将图片显示到界面。

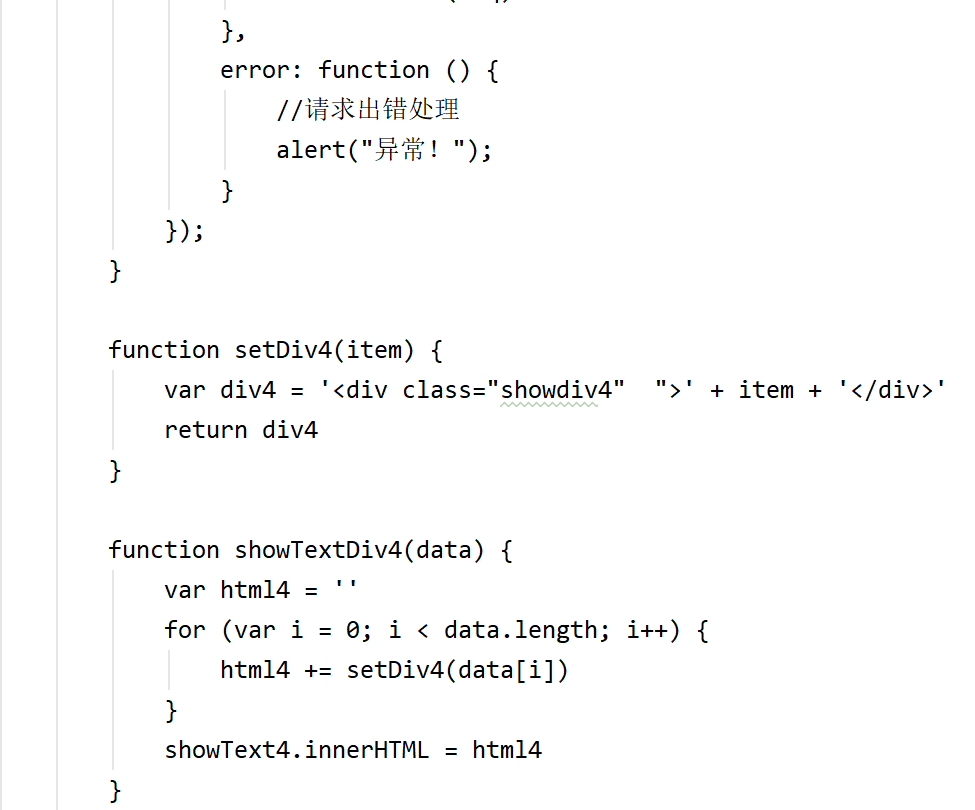
前端处理结果如图所示：



1. 接口部分：

代码展示如图（以 情感分类 为例，其余功能的接口均相似）





函数 sentiment() 作用是获取到相应功能的URL，然后进入submmit（）函数

函数 submmit() 的作用是异步请求和抛出异常，获取在后台与服务器进行少量数据交换，Ajax可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。同时抛出异常（异常通常发生在没有接收到前端传入数据时或后端返回值类型错误时）

函数 setDiv4()、 showTextDiv2() 作用是展示后端返回的处理结果。

1. 实验总结：

通过本次大作业，学会了使用flask框架，并能够结合bootstrap工具包构建出界面合理布局与样式。在第三次实验的基础上，我对“分词”、“词性标注”、“命名实体识别”、“情感分类”和“词云展示”等python语料处理功能有了更深入的理解，并能利用所学知识写出相对成熟的代码。在进行作业的过程中，我遇到了一些困难并加以克服，比如在编写代码期间，出现了jiagu包更新而不能用的情形，我及时上网查找了相关解决方案，并采取下载安装stanfordnlp包的方式进行替代，成功解决了问题。当然，我深知本次大作业还有好多地方需要改进，但是这个实验让我对本课程所学知识有了更深层次的理解和学习，也同时给予了利用所学的机会，锻炼了自学能力，为未来深入学习python和大数据相关知识奠定了重要基础。