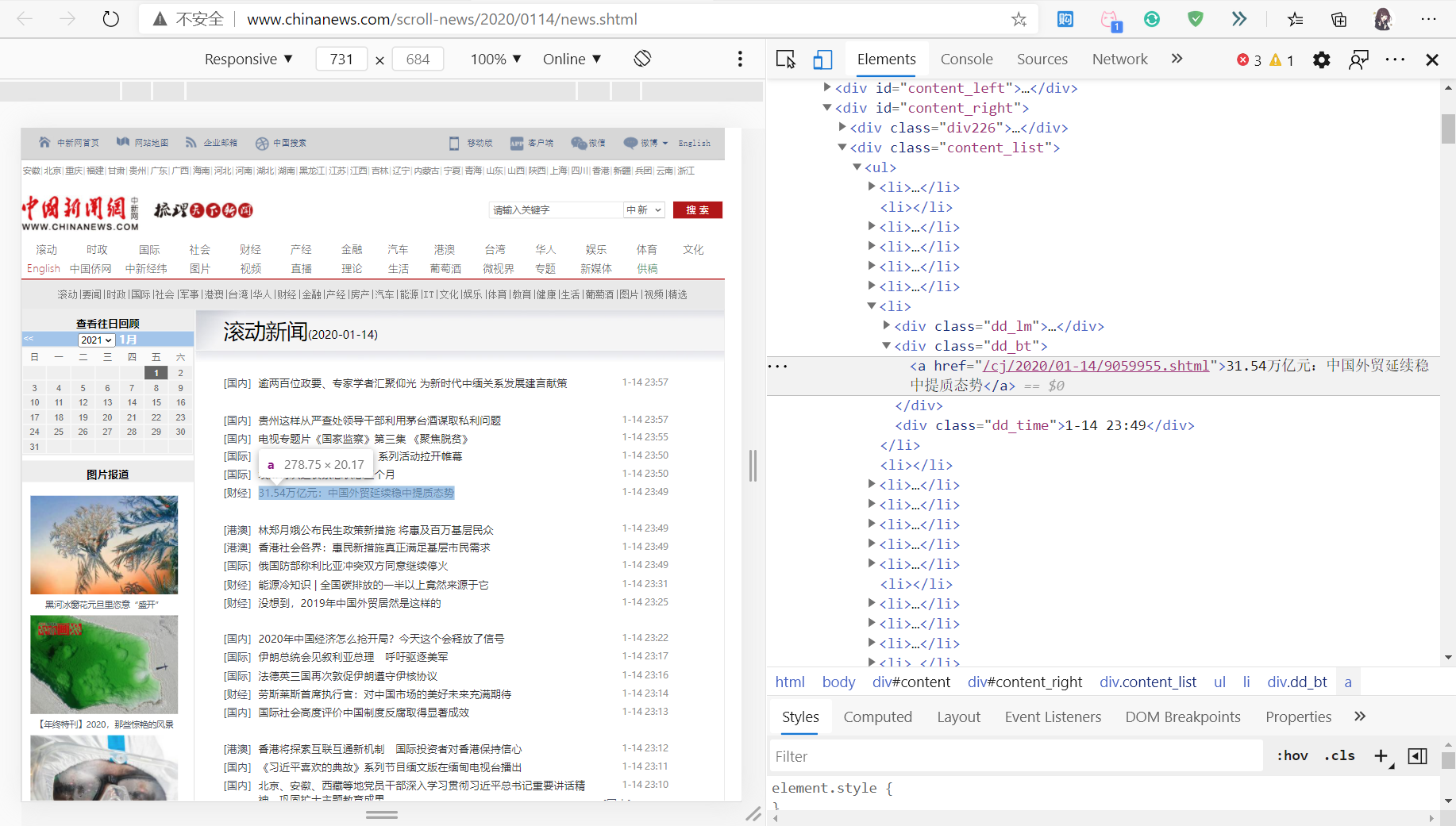
一、数据收集

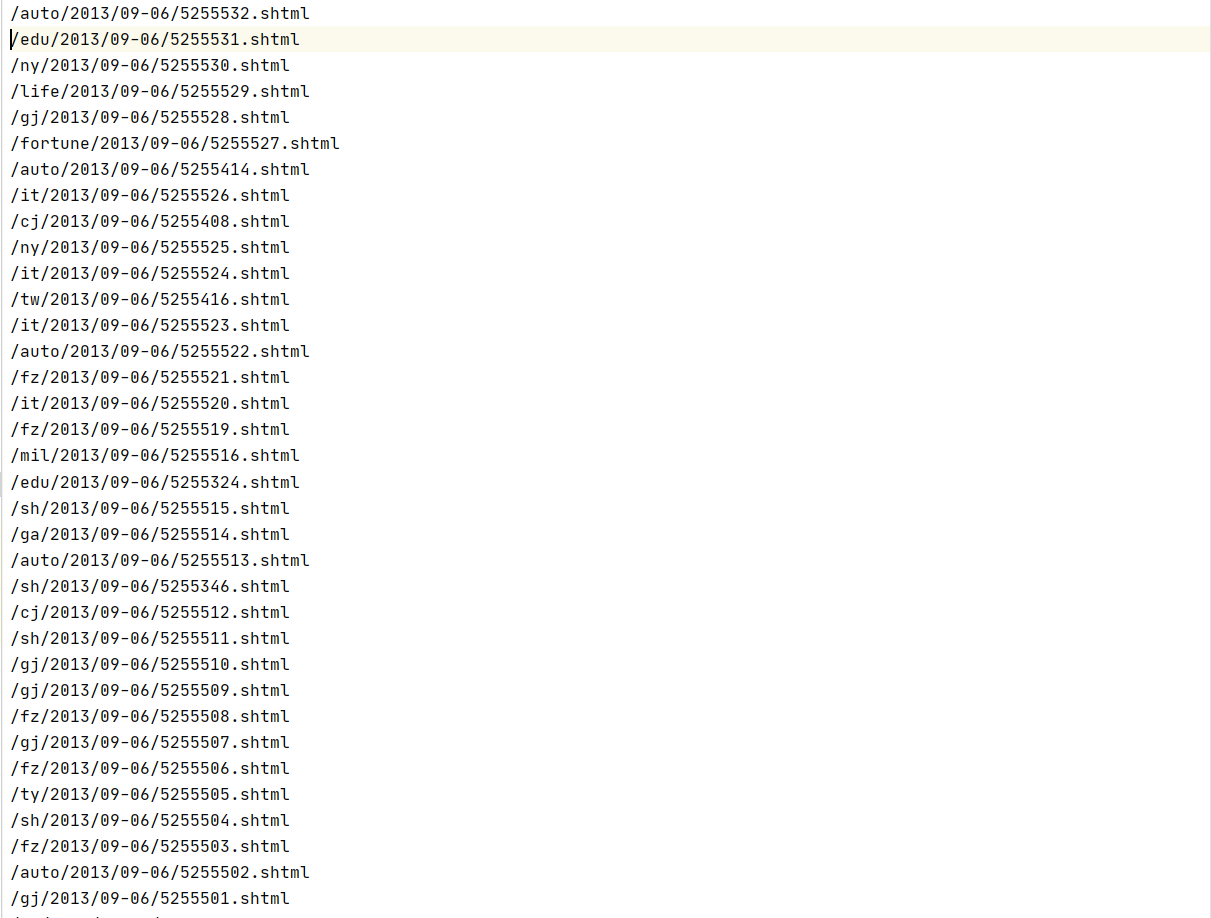
在本次实验中，我通过自编爬虫，获取中国新闻网的新闻文本。

第一步，在中国新闻网的“滚动新闻 （scroll-news）”页面，爬取所有新闻的url。其中，yyyy年mm月dd日的滚动新闻网页格式为<http://www.chinanews.com/scroll-news/yyyy/mmdd/news.shtml>。使用BeautifulSoup分析网页，提取出类别为‘dd\_bt’的div块，其中的内容为每条新闻网页url。





提取结果url.txt



第二步，使用收集到的新闻页面url，获取新闻文本，提取其中的’left\_zw’div块。特别要注意的是，2019年以前的网页使用GBK编码，而2019年后的网页使用utf-8编码。

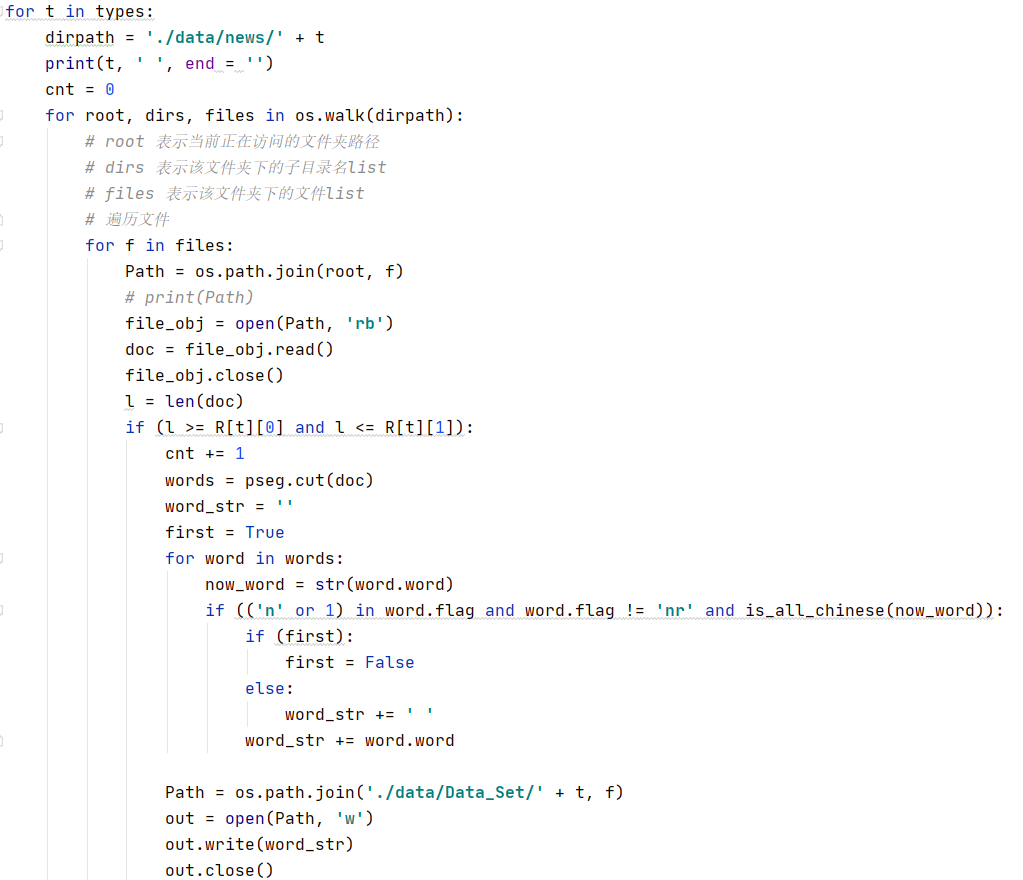


共提取200万条新闻。



二、数据预处理

首先在爬取到的新闻文本中，选择有代表性的10类，这里我选择['IT', '股票', '犯罪', '教育', '军事', '能源', '汽车', '体育', '娱乐', '住房']。在每一类中，筛选长度为2000~3000字的文章进行处理，从而使得数据集文档词数接近。



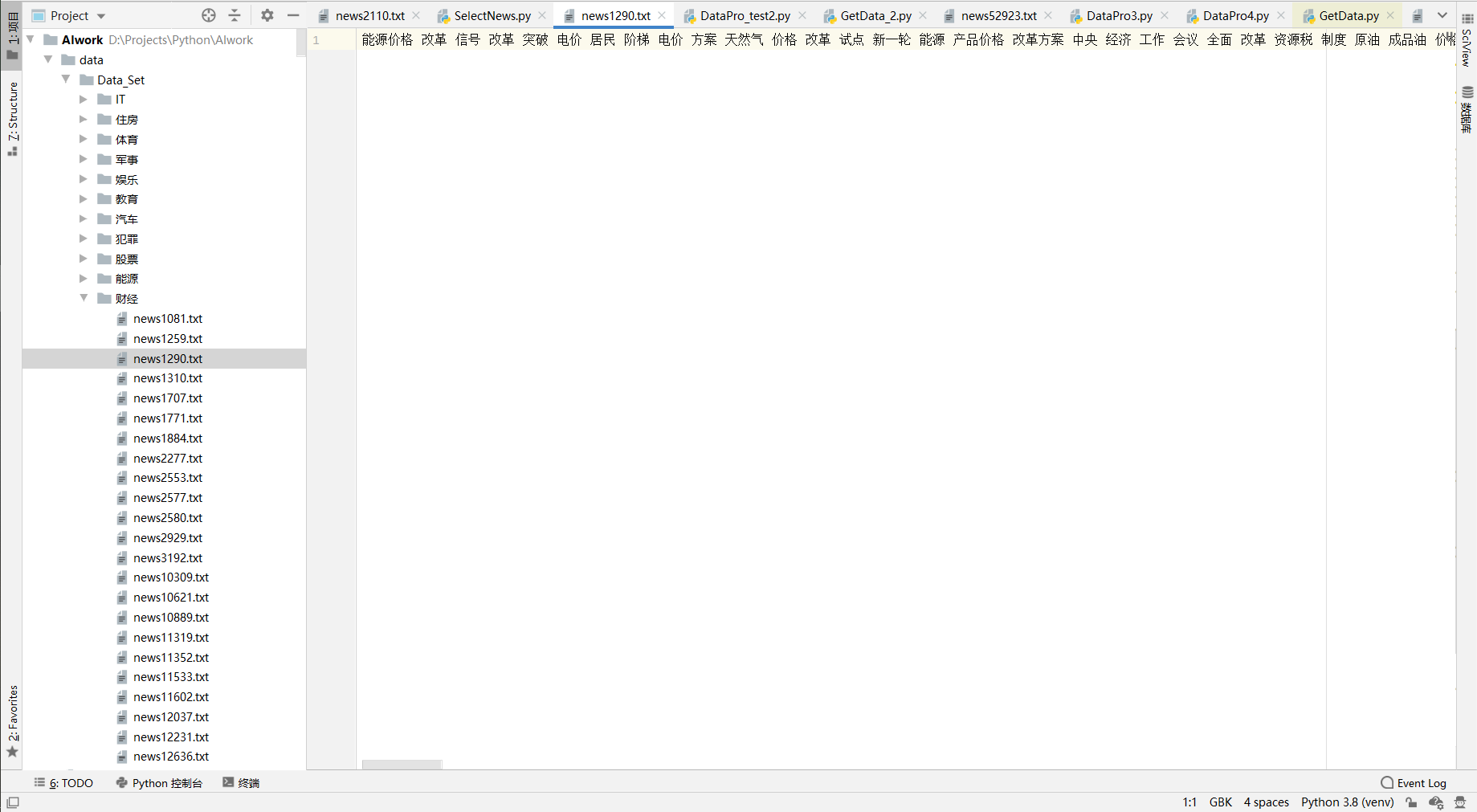
然后利用Jieba对筛选本文进行分词，分词后得到一个List，每个元素包含flag与word两个元素。其中flag表明词语的属性，如下表所示：

| **标签** | **含义** | **标签** | **含义** | **标签** | **含义** | **标签** | **含义** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | 普通名词 | f | 方位名词 | s | 处所名词 | t | 时间 |
| nr | 人名 | ns | 地名 | nt | 机构名 | nw | 作品名 |
| nz | 其他专名 | v | 普通动词 | vd | 动副词 | vn | 名动词 |
| a | 形容词 | ad | 副形词 | an | 名形词 | d | 副词 |
| m | 数量词 | q | 量词 | r | 代词 | p | 介词 |
| c | 连词 | u | 助词 | xc | 其他虚词 | w | 标点符号 |
| PER | 人名 | LOC | 地名 | ORG | 机构名 | TIME | 时间 |

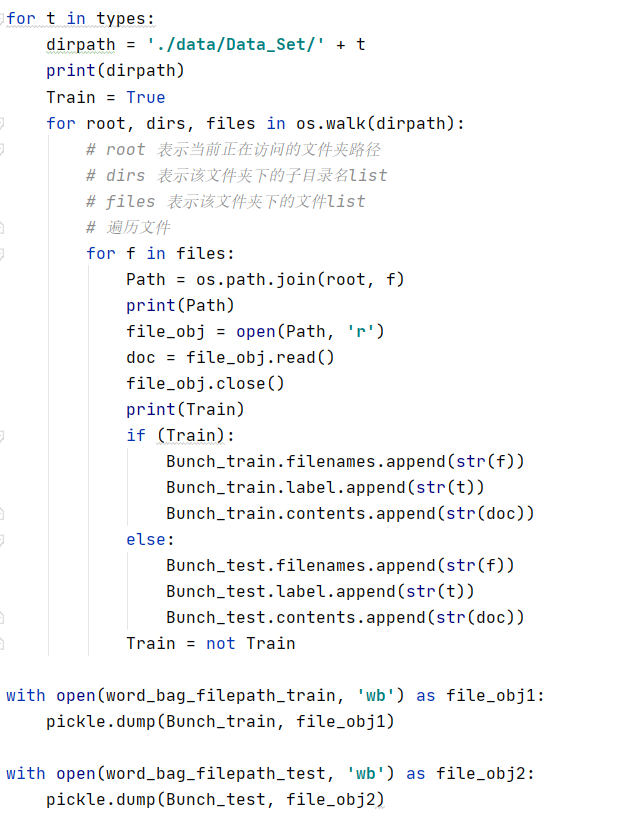
为了准确提取文章特征，我们只选择名词，且排除人名。



得到完成分词的10类文本，每类一万篇。



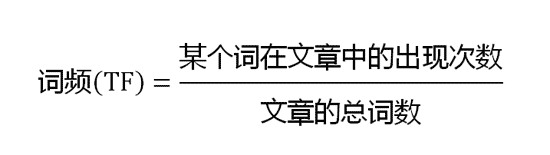
将每类的1万篇文档交替存入两个词袋模型，用于划分相等数量的训练集与测试集。

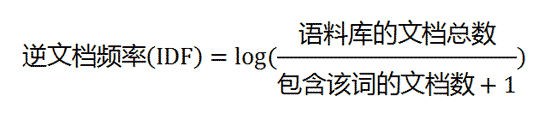


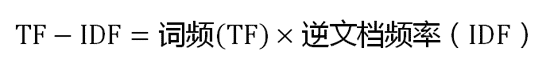
结果得到raw\_Bunch\_test与raw\_Bunch\_train两个原始词袋文件



使用TF-IDF模型提取训练集中的特征词。TFIDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。计算数据集中词频（TF）和逆文档频率（IDF），两者相乘得到TF-IDF。



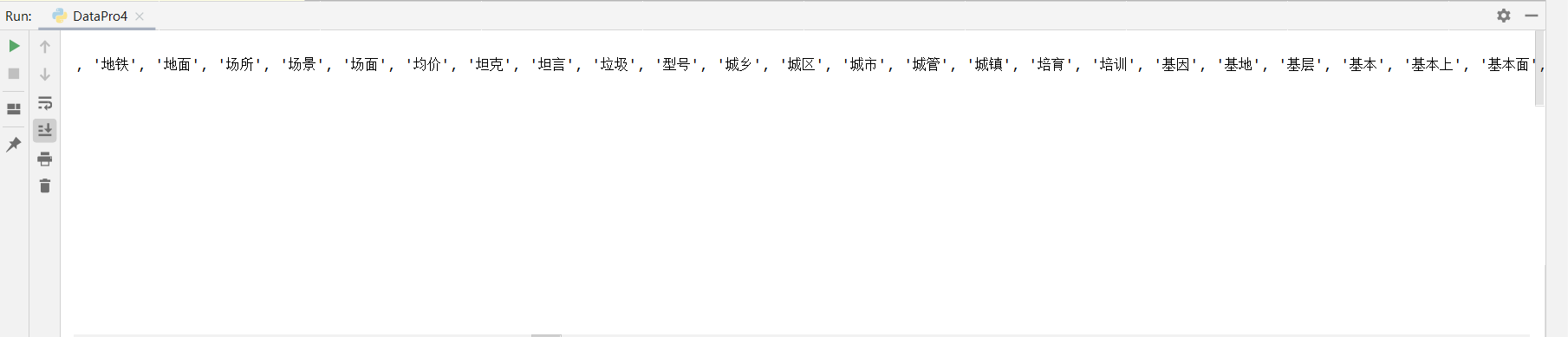




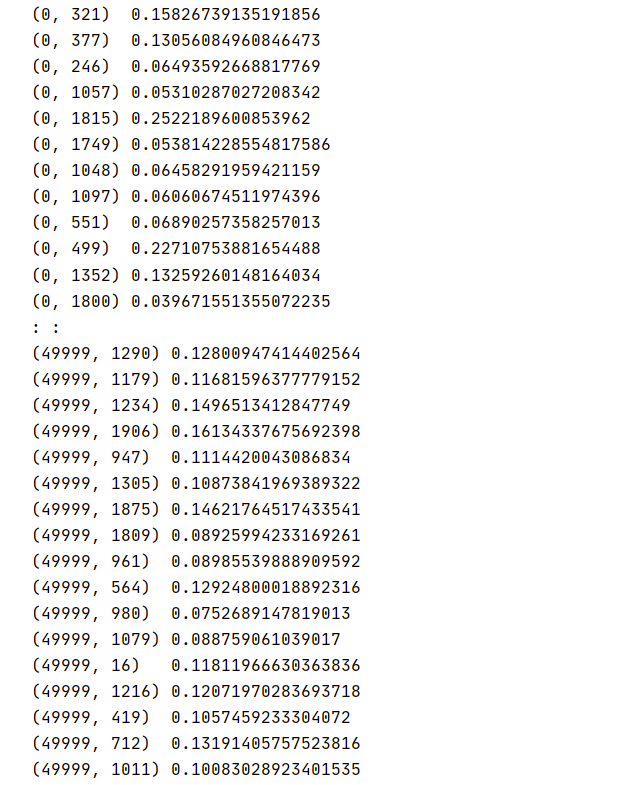
处理为TF-IDF模型后，利用TfidfVectorizer方法得到指定维度的字典为2000，得到数据字典，并生成训练集、测试集词袋模型。其中在生成TF-IDF的TfidfVectorizer指定去除在少于0.1%的文档中出现的词，在高于15%的文档中出现的词。本实验共选择10类文本，每类占比10%，如果一个词在15%以上的文档中都出现，说明它在不止一个类别中多次，那么该词语不具有好的代表性，应该去掉。对于少于0.1%的词，同样不具有代表性。



数据字典：



TF-IDF权重矩阵



（用numpy稀疏矩阵表示，(i,j)代表第i篇测试文档在第j维向量上的特征值）

用训练集得到的字典生成测试集的特征向量。

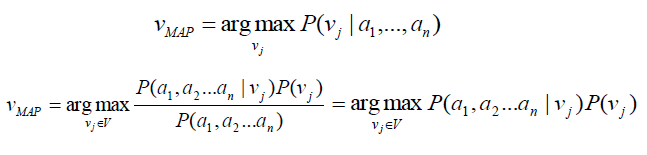


得到训练集与测试集的词袋文件



三、朴素贝叶斯分类器

利用贝叶斯公式，在给定描述实例的属性值，得到最可能的目标值。



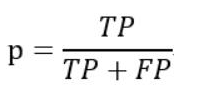
在此基础上，我们利用TF-IDF进行改进，使用TF-IDF将测试集中单词出现的次数转为一个0~1的TF-IDF权值，它在原贝叶斯方法的基础上额外考虑的单词的出现频率（重要程度）。与此同时取对数减少乘法运算次数，并在对数内加1以处理零概率。因此，计算后验概率式子变为



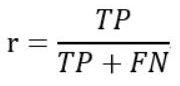
（为当前测试集第i个词的TF-IDF值）。

对于正确率、召回率、F1-score的计算。

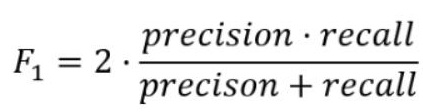
正确率（precision）指的是预测值为1且真实值也为1的样本在预测值为1的所有样本中所占的比例。



召回率（recall）指的是预测值为1且真实值也为1的样本在真实值为1的所有样本中所占的比例。

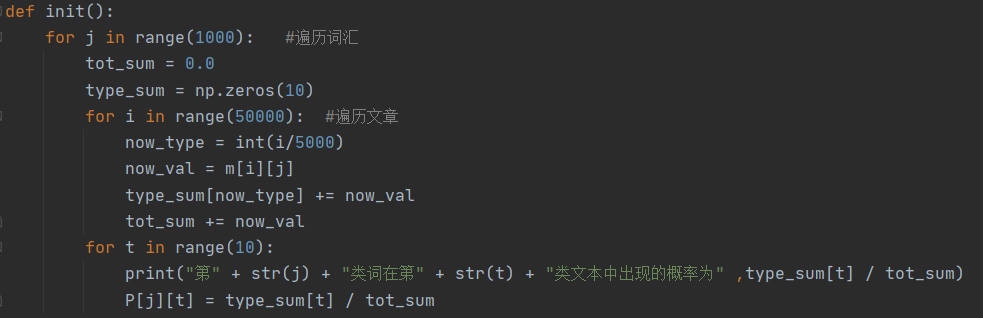


F1-score，又称为平衡F分数，是正确率和召回率的调和平均。



本实验的朴素贝叶斯分类器主要有三个步骤：

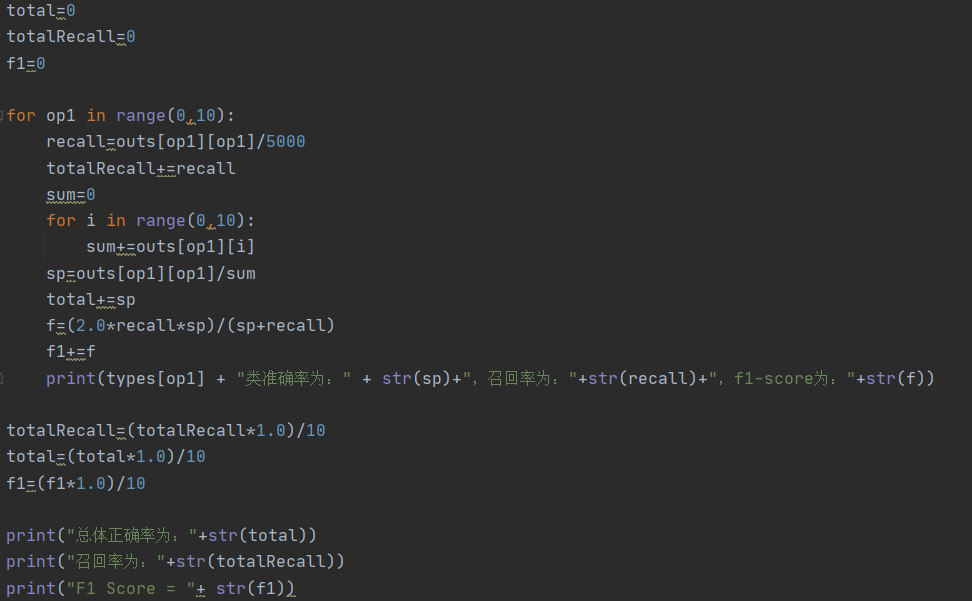
第一步，通过前面预处理后的文本数据，训练得到在各个类别中出现各个特征词汇的概率，同时得到各类文章的统计概率；



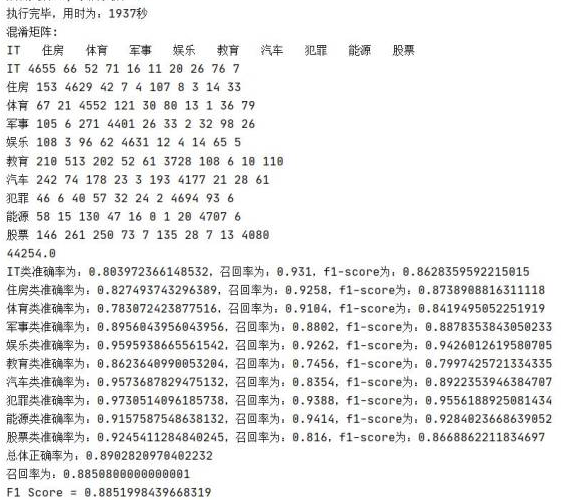
第二步，对于测试集的每一篇文章遍历，然后对每一篇文章的每一个词遍历，去找这个词在训练集中的概率，也就是对训练集中所有该列，分10类遍历，求最大。



第三步，打印混淆矩阵，计算各类和总体正确率、召回率、F1-score。



实验结果：平均正确率达到89.03%，其中最高准确率可达97.31%。



四、SVM分类器

这里本实验采用的是libsvm的LinearSVC。

首先介绍一下svm。支持向量机（Support Vector Machine, SVM）是一类按监督学习（supervised learning）方式对数据进行二元分类的广义线性分类器（generalized linear classifier），其决策边界是对学习样本求解的最大边距超平面（maximum-margin hyperplane）。SVM使用铰链损失函数（hinge loss）计算经验风险（empirical risk）并在求解系统中加入了正则化项以优化结构风险（structural risk），是一个具有稀疏性和稳健性的分类器。SVM可以通过核方法（kernel method）进行非线性分类，是常见的核学习（kernel learning）方法之一。

接着介绍一下libsvm。libsvm是台湾大学林智仁(Lin Chih-Jen)教授等开发设计的一个简单、易于使用和快速有效的SVM模式识别与回归的软件包，他不但提供了编译好的可在Windows系列系统的执行文件，还提供了源代码，方便改进、修改以及在其它操作系统上应用；该软件对SVM所涉及的参数调节相对比较少，提供了很多的默认参数，利用这些默认参数可以解决很多问题；并提供了交互检验(Cross Validation)的功能。该软件可以解决C-SVM、ν-SVM、ε-SVR和ν-SVR等问题，包括基于一对一算法的多类模式识别问题。

而支持向量分类的SVC（C-Support Vector Classification）就是基于libsvm实现的。其数据拟合的时间复杂度是数据样本的二次方，这使得他很难扩展到10000个数据集，当输入是多类别时（SVM最初是处理二分类问题的），通过一对一的方案解决。

LinearSVC（Linear Support Vector Classification）：线性支持向量分类，类似于SVC，但是其使用的核函数是”linear“上边介绍的两种是按照brf（径向基函数计算的，其实现也不是基于LIBSVM，所以它具有更大的灵活性在选择处罚和损失函数时，而且可以适应更大的数据集，他支持密集和稀疏的输入是通过一对一的方式解决的。

LinearSVC 参数解释

C：目标函数的惩罚系数C，用来平衡分类间隔margin和错分样本的，default C = 1.0；

loss ：指定损失函数

penalty ：

dual ：选择算法来解决对偶或原始优化问题。当n\_samples > n\_features 时dual=false。

tol ：（default = 1e - 3）: svm结束标准的精度;

multi\_class：如果y输出类别包含多类，用来确定多类策略， ovr表示一对多，“crammer\_singer”优化所有类别的一个共同的目标

如果选择“crammer\_singer”，损失、惩罚和优化将会被被忽略。

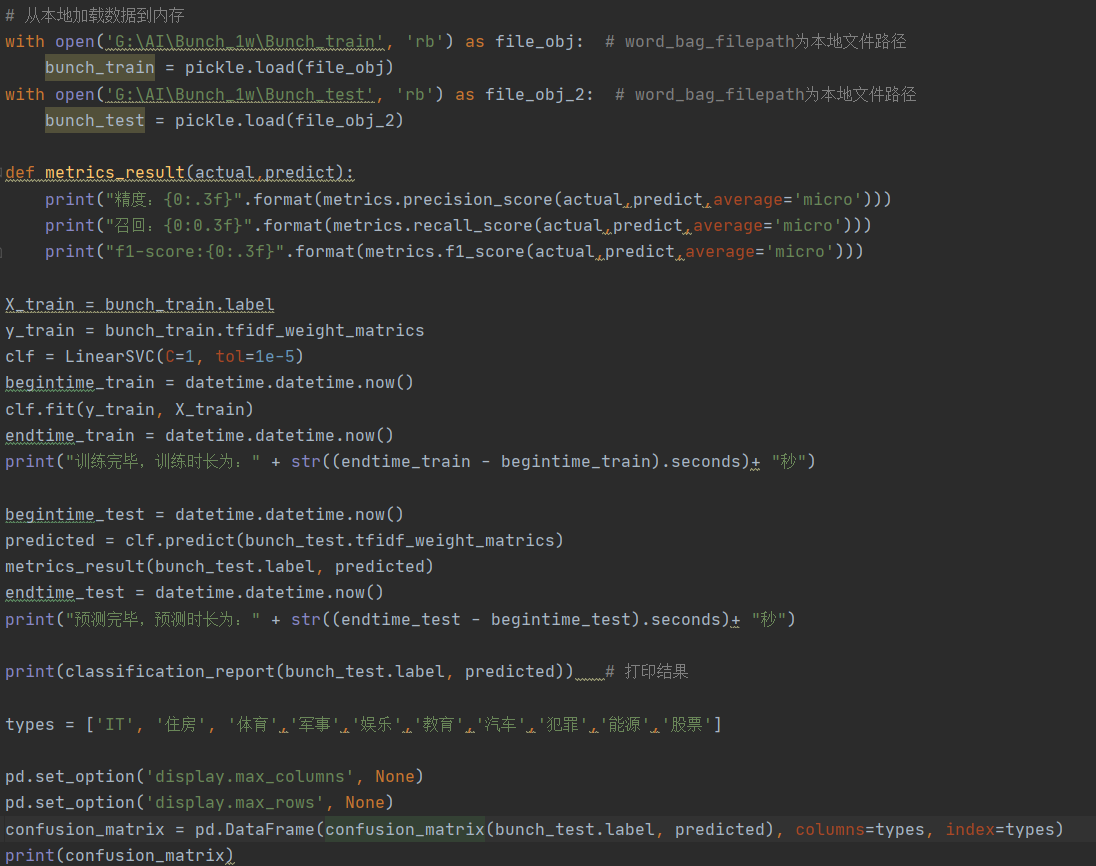
fit\_intercept ：

intercept\_scaling ：

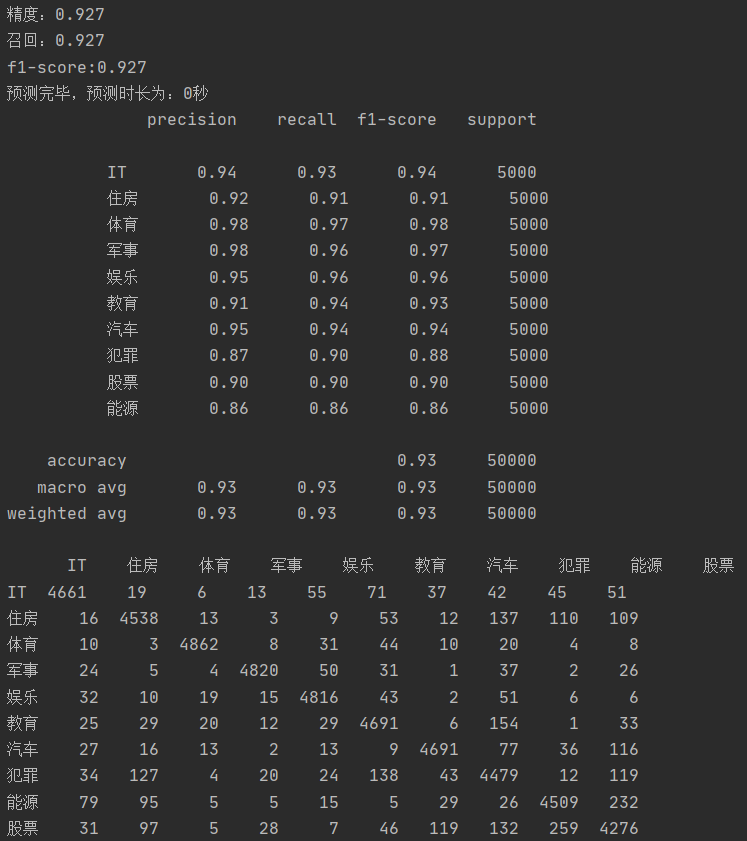
class\_weight ：对于每一个类别i设置惩罚系数C = class\_weight[i]\*C,如果不给出，权重自动调整为 n\_samples / (n\_classes \* np.bincount(y))

verbose：跟多线程有关

在整个实验中，所有的训练、测试、打印混淆矩阵和正确率等参数的过程都可以通过调用sklearn.svm中的函数实现。



实验结果：平均正确率达到92.7%，其中最高准确率可达98%。



五、思考与体会  
胡天翼：

熊宇：在本次文本分类实验中，我对于朴素贝叶斯的原理及实现过程有了更加深刻的理解，在编程实现朴素贝叶斯的过程中，通过对其进行相关改进的尝试，体会到了数据预处理对于实验繁杂程度及实验结果的损失程度的影响，认识到了人工智能对于数据的依赖能力；在调包实现SVM的过程中，通过对比朴素贝叶斯和libsvm，了解了不同分类器的不同思路，培养了对人工智能、数据科学的兴趣，提高了学习、汲取、进步的能力，受益匪浅。