**作业2-1**

分别收集尽量多的英语和汉语文本，编写程序计算这些文本中英语字母和汉字的熵，对比本章课件第12页上表中给出的结果。然后逐步扩大文本规模，如每次增加2M，重新计算文本规模扩大之后的熵，分析多次增加之后熵的变化情况。

要求：

1. 利用爬虫工具从互联网上收集样本，并对样本进行处理，如清洗乱码等；
2. 设计算法并编程实现在收集样本上字母/汉字的概率和熵的计算；
3. 当改变样本规模时，重新计算字母/汉字的概率和熵的计算，并分析计算结果；
4. 完成一份技术报告，在报告中写明利用什么爬虫工具从哪些网站上收集的样本，如何进行的样本清洗，清洗后样本的规模，在不同样本规模下计算的结果等。

第一步：利用爬虫爬取英语和汉语的文本

本练习使用python语言进行编程，requests库用来获取HTML网页，并用parsel库对网页进行解析，并从不同网站上获取英语和汉语文本。

**1. 英语文本爬取**

（1）网站

英文小说网(<http://novel.tingroom.com/>)

（2）文本爬取

主要分为2步：①访问每部小说的url并获取每一章小说的url；②访问每一章的url并获取相应的英文文本信息进行保存。代码实现见附件。

（3）爬取结果

由于需要尽可能多的文本数据，所以就对多部小说进行了爬取，最终共爬取了约146部英文小说，文件大小约为47.6M。

**2. 汉语文本爬取**

（1）网站

蚂蚁文学(<http://www.biquges.com/>)

（2）文本爬取

主要分为2步：①访问每部小说的url并获取每一章小说的url；②访问每一章的url并获取相应的中文文本信息进行保存。代码实现见附件。

（3）爬取结果

最终爬取约10部中文小说，文件大小约为46.5M。

在收集完英文和中文文本之后，会发现其中除了含有英文字母和汉字以外，还有其他的字符和乱码，所以接下来需要进行数据的清洗。

第二步：对爬取到的文本进行清洗

**1. 英语文本清洗**

图1为爬取到的原始英文文本，可以看到其中除了含有英语字母之外，还有多余的空白、标点符号、数字等其他字符，并且还有大写和小写形式，需要通过英文文本清洗对原始数据进行处理。

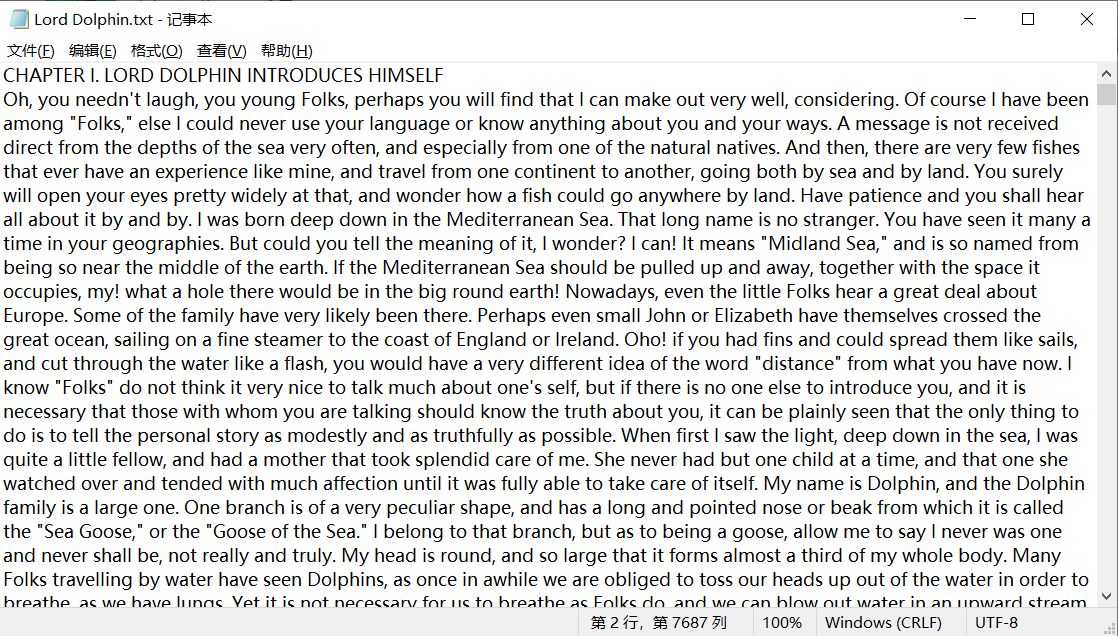


图1 原始英文文本

对于英文文本的清洗分为5步：①去除多余空白、分词，并进行词性标注；②对词形进行还原；③删除停用词；④删除特殊字符、数字；⑤将大写转为小写。

对于英文文本数据清洗本作业借助了Python中的自然语言处理库NLTK模块，它可以将段落拆分为句子、拆分词语、识别词性等等。

（1）去除多余空白、分词，并进行词性标注

去除多余空白：将多余的空白替换成一个空格；

分词：利用NLTK包中的word\_tokenize()方法对每个单词进行划分；

词性标注：利用NLTK包中的pos\_tag()方法给每一个词打上词性标签，例如动词(VB)、名词(NN)、形容词(JJ)等；

（2）对词形进行还原

对每一个词性进行标注后，接着利用WordNetLemmatizer包中的lemmatize()方法对词形进行还原，例如men->man, learning->learn等等；

（3）删除停用词

对词形还原后，需要对英语中的停用词删除，因为这些词不会给文本增加太多信息，利用NLTK中的stopwords包进行删除，停用词如：the, is, are等等；

（4）删除特殊字符、数字

接着利用正则表达式对文本中多余的字符和数字进行删除，例如：”, . < \* # 0 1 2 3”等，处理后的文本只剩下英文大小写字母；

（5）将大写转为小写

最后是将文本中的大写字母都转为小写字母并进行存储，最终清洗后的英文文本如图2所示，清洗后的英文文本规模为23.7M。



图2 清洗后的英文文本

**2. 汉语文本清洗**

图3为爬取到的原始汉语文本，可以看到其中除了含有汉字之外，还有多余的空白、标点符号、数字等其他字符，需要通过汉语文本清洗对原始数据进行处理。

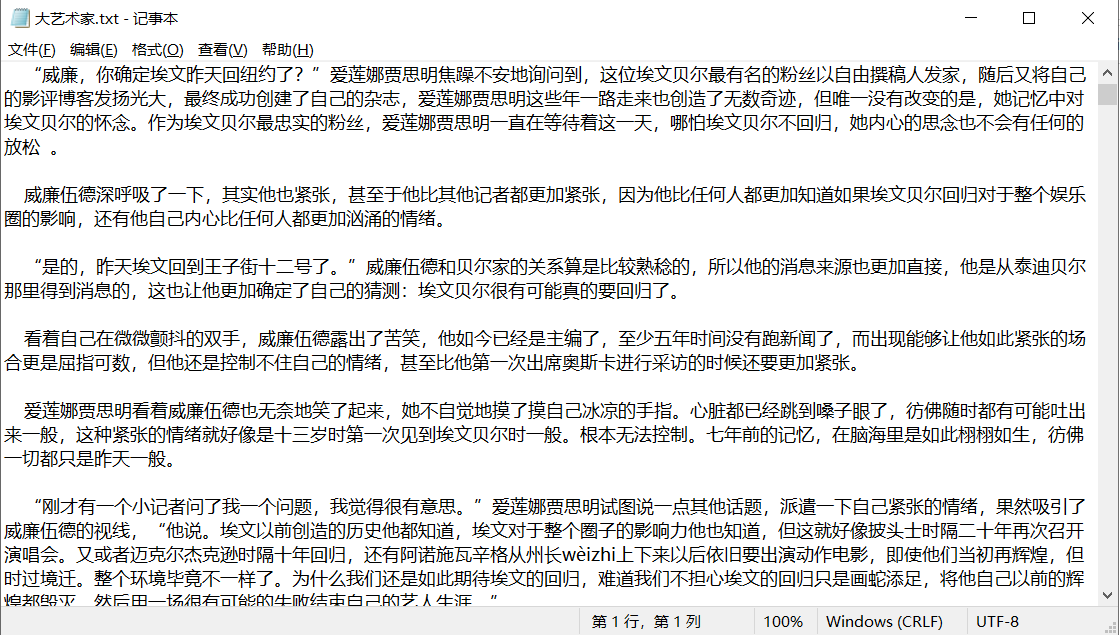


图3 原始汉语文本

对于汉语文本的清洗分为3步：①删除多余字符，提取所有汉字；②分词；③删除停用词。借助了正则表达式和jieba中文分词库等工具完成文本清洗工作。

（1）删除多余字符，提取所有汉字

利用正则表达式：’[^\u4e00-\u9fa5]’可以将非汉字提取，包括标点符号、数字等，并将其删除；

（2）分词

利用jieba中文分析库，将一个句子分成对应的词语，便于后续删除停用词；

（3）删除停用词

常用的停用词表有中文停用词表(cn\_stopwords.txt)、哈工大停用词表(hit\_stopwords.txt)、百度停用词表(baidu\_stopwords.txt)、机器智能实验室停用词表(scu\_stopwords.txt)等，本实验使用的是哈工大停用词表，停用词包括：“而且、不如、除了”等等。将这些停用词删除。清洗后的汉语文本如图4所示，清洗后的汉语文本规模为26.6M。

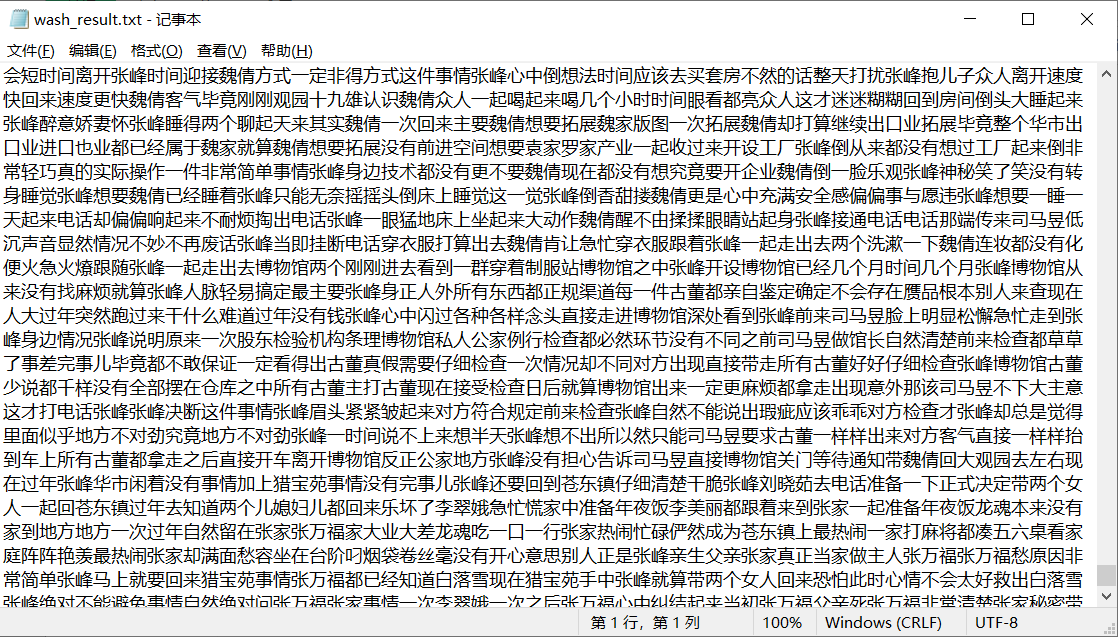


图4 清洗后的汉语文本

第三步：对清洗后的文本计算字符出现的概率/熵

**1．英语文本的熵**

对于英语文本，我们将清洗后的文件(23.7M)分为了10个子文件，每个文件大小为2M左右，每次往实验数据中加入一个子文件，逐步扩大文本的规模并计算英文字母的熵，利用熵的计算公式计算并观察熵的变化情况，实验过程中统计字母数和总的字母个数，实验结果如表1所示。

表1 英语文本的熵统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文本 | 规模 | 字母数 | 总字母个数 | 熵 |
| text1 | 2.09M | 26 | 2199596 | 4.2341 |
| text2 | 3.94M | 26 | 4132880 | 4.2388 |
| text3 | 6.15M | 26 | 6451353 | 4.2353 |
| text4 | 8.49M | 26 | 8903941 | 4.2360 |
| text5 | 10.6M | 26 | 11123708 | 4.2355 |
| text6 | 13.7M | 26 | 14368997 | 4.2363 |
| text7 | 15.6M | 26 | 16434857 | 4.2368 |
| text8 | 18.3M | 26 | 19194567 | 4.2375 |
| text9 | 20.7M | 26 | 21717242 | 4.2371 |
| text10 | 23.7M | 26 | 24843920 | 4.2366 |

根据每次实验计算出的英语字母的熵，绘制出英语字母熵的变化折线图如图5所示。

图5 英语字母熵变化折线图

由实验结果可知，随着英语文本规模的逐渐增加，英语字母的熵也在发生细微的波动，由于英语字母总共只有26个，所以熵的变化不大，10次实验的英语字母熵的均值约为4.2364。

**2. 汉语文本的熵**

对于汉语文本，我们将清洗后的文件(26.6M)分为了10个子文件，每个文件大小为2M左右，每次往实验数据中加入一个子文件，逐步扩大文本的规模并计算汉字的熵，利用熵的计算公式计算并观察熵的变化情况，实验过程中统计汉字数和总的汉字个数，实验结果如表2所示。

表2 汉语文本的熵统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文本 | 规模 | 汉字数 | 总汉字个数 | 熵 |
| text1 | 2.06M | 3424 | 723131 | 9.2445 |
| text2 | 4.46M | 4332 | 1561812 | 9.7885 |
| text3 | 6.72M | 4479 | 2349128 | 9.9078 |
| text4 | 10.9M | 4686 | 3838532 | 10.0059 |
| text5 | 14.5M | 4760 | 5094409 | 9.9365 |
| text6 | 17.6M | 4947 | 6178989 | 9.9832 |
| text7 | 19.2M | 4967 | 6721661 | 9.9994 |
| text8 | 21.0M | 5080 | 7359332 | 10.0042 |
| text9 | 24.1M | 5198 | 8446421 | 10.0044 |
| text10 | 26.6M | 5209 | 9330563 | 9.9731 |

根据每次实验计算出的汉字的熵，绘制出汉字熵的变化折线图如图6所示。

图6 汉字熵变化折线图

由实验结果可知，随着汉语文本规模的增加，其中的汉字数也在不在增加，在第一次实验中，由于文本规模较小，且仅有3424个汉字，计算出的汉字的熵为9.2445，当不断增加文本规模，可以看到汉字的熵逐渐增大，最终在10左右波动，10次实验的英语字母熵的均值约为9.8848。

注：

代码和数据分别见附件1和附件2。

**附件1：代码**

其中包含8个.py文件，文件名及其作用见下表：

|  |  |
| --- | --- |
| **文件名** | **作用** |
| Crawl\_English\_Text.py | 爬取英语文本 |
| Crawl\_Chinese\_Text.py | 爬取汉语文本 |
| Clean\_English\_Text.py | 清洗英语文本 |
| Clean\_Chinese\_Text.py | 清洗汉语文本 |
| Cal\_English\_Entropy.py | 计算英语字母的熵 |
| Cal\_Chinese\_Entropy.py | 计算汉字的熵 |
| Merge\_TXT.py | 合并.txt文件 |
| Split\_TXT.py | 分割.txt文件 |

**附件2：数据**

其中包含英语和汉语清洗并分割后的文本数据，各10个子文件，由于全部的数据量较大，将把全部数据放在github中，链接为：