技术报告

2019K8009907018 王畅路

问题分析

问题的描述为：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。
根据问题得到以下需求和解决方案：

- 收集大量自然语言文本　　--使用爬虫工具收集
- 对收集到数据进行处理　　--使用正则表达式工具筛选数据
- 改变样本规模计算熵　　--使用python编程计算熵

爬取数据

为了得到大量的文本数据，我选择了人民日报的官方网站人民日报-人民网(people.com.cn)

第01版:要闻　　PDF下载

在爬虫工具的选择上，使用了scrapy框架

1.为了爬取每天人民日报的数据，每次爬取需要对官网的url进行处理，把其中涉及日期的部分更改为要爬取的日期。

```
    url = self.base_urls + self.parseday.isoformat()[:-3] + "/" + self.parseday.isoformat()[8:10]+"/nbs.D110000renmrb_01.htm"
    yield scrapy.Request(url, self.parse)
```

2.由于每天的日报会分成不同的模块，因此需要对每个模块的url发出http请求

```
        yield scrapy.Request(page_links+"nbs.D110000renmrb_01.htm",callback=self.parse)
```

3.每个模块中又会有不同种类的文章，因此需要对每个文章的url发出http请求

```
        yield scrapy.Request(next_passage,callback=self.parsefinal)
```

4.得到了请求的response之后，可以对html进行解析，使用css选择器，最后就可以得到想要的数据

```
            yield {
                'title': article.css('h1::text').get(),
                'author': article.css('p.sec::text').get(),
                'contents': article.css('div#ozoom p::text').getall(),
            }
```

5.最后使用命令行命令爬取 `scrapy crawl spider -O renmin.json`，把爬取到的数据爬到json文件中

由于人民日报仅仅提供了2021.1.1~至今的文章，最后总共爬取了30623篇文章

数据处理

由于爬取的数据包含了标点符号等无用信息，因此需要对数据进行筛选。
选择使用python自带的json处理库和re正则表达库
由于在unicode的编码中 中文的编码范围是[\u4e00-\u9fff]，因此只需要保留这部分数据即可
1.首先把数据从json文件中加载出来
2.解析到需要的部分

3.保留字符串中的中文字符

4.拼接各个段落

5.把得到的数据存到新的json文件中

经过数据处理后，得到的文件大小为125.42MB,总字数为40298382

计算熵

分为两个步骤：

1.增加样本规模

我选择了以2000篇文章为单位来逐步增加样本规模。每次增加规模会给出当前的样本量（单位：个）

增加规模的方式如下

```
for i in range(15):#15次迭代
    for j in range(2000):#每次处理2000篇文章
        processing_index = j + i * 2000
        processing_string = json_data[processing_index]['result']#处理的串（文章）
```

2.计算熵

根据计算熵的公式

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

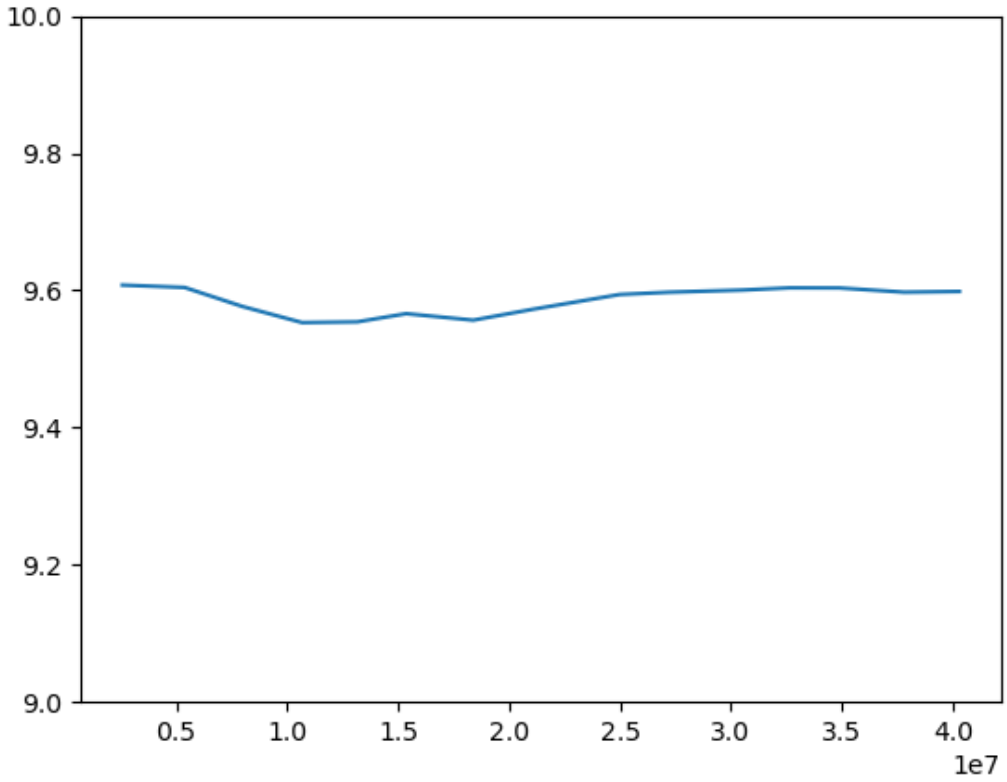
具体计算熵的代码如下

```
all_likelihood = {}#概率字典，记录每个字的概率
for k in all_character.keys():
    all_likelihood[k] = all_character[k]/character_number#计算每个字的概率
entropy = 0
for k in all_likelihood.keys():
    single = (all_likelihood[k]) * math.log2(all_likelihood[k])#根据公式计算每个字的熵
    entropy = entropy - single#求和
print("text length:",character_number,"entropy:",entropy,"\n",end='')
```

获得的15次结果如下

```
time 1 text length: 2567886  entropy: 9.607675732847914
time 2 text length: 5375363  entropy: 9.604189354303225
time 3 text length: 7990078  entropy: 9.576397206672098
time 4 text length: 10672930  entropy: 9.553000906434429
time 5 text length: 13160365  entropy: 9.554074659728464
time 6 text length: 15364742  entropy: 9.565983896137416
time 7 text length: 18385490  entropy: 9.556539792421944
time 8 text length: 21082701  entropy: 9.572472497998326
time 9 text length: 24983840  entropy: 9.593930089129865
time 10 text length: 27196462  entropy: 9.597241055806276
time 11 text length: 30514819  entropy: 9.600407711084396
time 12 text length: 32650823  entropy: 9.6036097864512
time 13 text length: 34894112  entropy: 9.603457355033829
time 14 text length: 37813677  entropy: 9.597315243092162
time 15 text length: 40298382  entropy: 9.598337642007174
```

使用python进行数据可视化



可以发现，不同规模下，中文汉字的熵稳定在9.6附近，随着样本增大不断平滑。

查资料可知，汉字的静态平均信息熵是9.65比特，和所得结果相近。

汉字的静态熵比很多语言都要高，这意味着中文信息处理的难道度要大于比这些语言。