自然语言处理技术报告

姓名：袁昭新

学号：2020K8009926029

一、数据来源

为了对比不同样本的差别，中文样本分别在 维基百科、纵横中文网 以及 人民网 上收集；英文样本在维基百科、FullEnglishBooks、ABC News 以及 CNN 上收集。样本的类型覆盖了网络百科全书、小说和网络新闻。

图 1-数据来源

为了使数据尽可能多样，在爬取维基百科选取了多个类别作为起点，例如：科学、社会、文化、自然、宗教等，并在其中进行递归搜索。其他网站则遍历首页的文章，并在文章的相关推荐中递归爬取。

二、爬虫工具

Python 提供了 requests 模块，用于发送 HTTP 请求，然后利用 Beautiful Soup 库从 HTML 文件中解析和提取数据。基本的使用方式如下：

```
import requests
from bs4 import BeautifulSoup

response = requests.get(url)
html_doc = response.text
soup = BeautifulSoup(html_doc, 'html.parser')
```

根据不同网站的特点，使用不同的方式来爬取足够数量的文本。例如，维基百科的每个词条中都含有大量指向其他词条的链接；对于 CNN，含有大量文本的网页一般都以"index.html"结尾；对于小说网站，每一页的结尾都有指向下一页的链接……可以通过递归或 BFS 的方式来获取足量的文本。

另外，为了防止爬取到重复的内容，使用一个列表来存储待爬取的链接，每次发现新的链接时都需要检查是否已经在列表中。

递归的主函数大致如下（根据网站略有不同）：

```
def crawl(current_url, file):
    print(f'正在访问链接: {current_url}')
    # 获取当前链接对应的网页内容
    ......
    # 写入文件
    wr_txt(file, bodytxt)
    try:
        # 获取小说下一章的链接
        if href:
            crawl(href, file)
    except:
        pass
```

BFS 的主函数大致如下（根据网站略有不同）：

```
def bfs(url, file):
    queue = [] # 存储待访问的链接
    counter = 0  # 记录已经爬取的次数
    while queue:
        if counter >= max:
            break
        # 超过最大数量则停止
        current_url = queue.pop(0)
        if current_url not in visited:
            try:
                visited.add(current_url)
                # 获取内容
                ......
                if souptext:
                    # 写入文件
                if len(queue) < max:
                    # 获取当前链接中的所有链接
                    ......
                    # 将符合条件的链接加入列表
            except:
                pass
```

三、数据处理

获取的文本中往往含有乱码、数字等不需要的东西，使用 Python 来清洗样本，仅保留中文或者英文字符。

对于中文样本，除了非中文外，从维基百科上爬取的内容可能还含有繁体字，因此使用 opencc 库先将繁体字转化为简体字，再使用正则表达式去除所有非中文字符。汉字的 unicode 编码范围为 4e00 到 9fa5，因此样本清洗方法如下:

```
# 创建OpenCC对象，指定转换规则
converter = opencc.OpenCC('t2s')
# 删除任何非汉字符号,定义正则表达式，匹配除了中文以外的任何字符
pattern = re.compile(r'[^\u4e00-\u9fa5]')
# 逐行读取输入文件内容，并将每行繁体字转换为简体字，然后写入输出文件
for line in input_file:
    simplified_line = converter.convert(line.strip())
    cleaned_line = re.sub(pattern, '', simplified_line.strip())
    output_file.write(cleaned_line)
```

对于英文样本，保留其中的英文和空格（同时将多个空格合并为一个），并将大写字母转为小写：

```
# 删除非字母
text = re.sub(r'[^a-zA-Z\s]', '', text)
text = re.sub(r"\s+", " ", text).strip()
# 大写转小写
text = text.lower()
```

清洗前后的样本规模如下表所示：

来源	原始样本规模(MB)	清洗后规模(MB)	丢弃率
维基百科(英文)	134.3	126.5	5.81%
ABC News & CNN	43.4	41	5.53%
Norvels	147.7	136.8	7.38%
维基百科(中文)	68.6	48.1	29.88%
人民网	69.2	60.3	12.86%
纵横中文网	62.6	53.2	15.02%

表1

四、数据分析

（一）不同样本的熵

每次添加 1M 个英文（或中文）字母，计算文本的熵，并绘图如下：

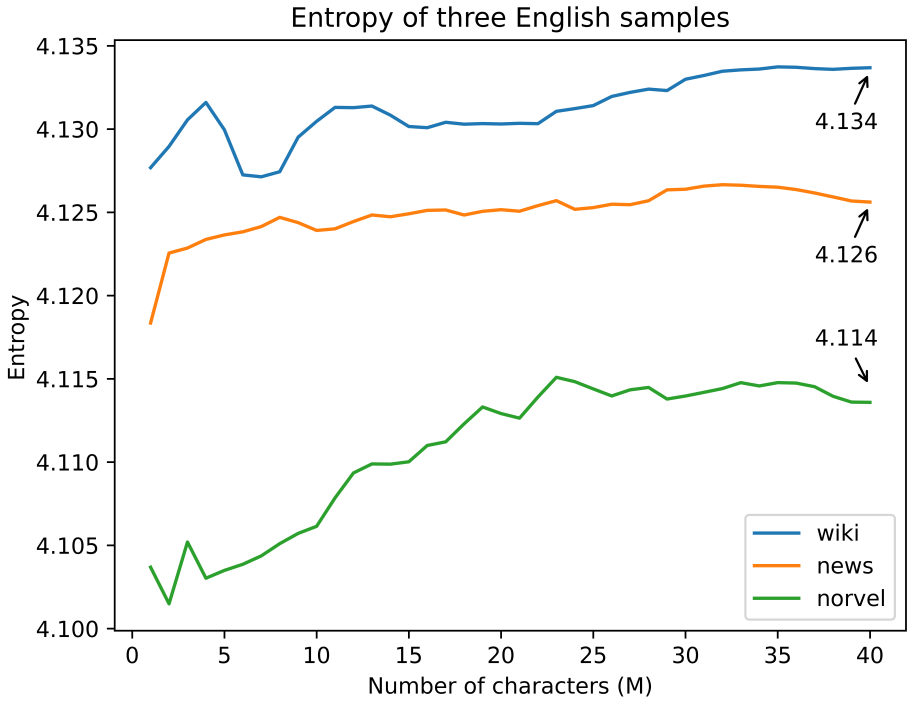

图 2-三个英文样本的熵

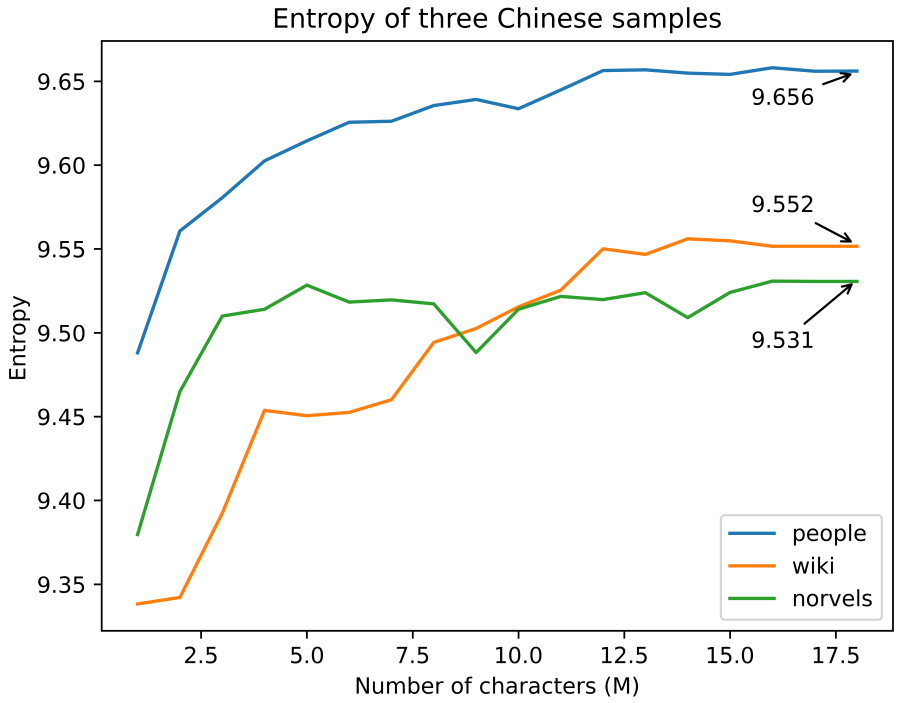

图 3-三个中文样本的熵

随着样本规模的增大，样本的熵逐渐趋于稳定。对于英文而言，三种不同来源的样本的熵有一定的差别，并且这种差别较为稳定。而对于中文而言，三种来源的样本之间熵的差距比英文更明显，而且更不稳定。

在中英文样本中，当样本达到一定的规模后，熵最低的都是小说。小说的特点决定了其中会出现重复的人名、地名等，所以其熵可能因此偏低。

（二）中、英文的熵

将样本合并之后计算熵，结果如下：

中文熵	英文熵
9.83036	4.13069

表2

合并所有文本后中文的熵变大了，这可能是因为三个样本的重合程度稍低造成的。另外，观察统计结果发现，样本中生僻的字较多，这也可能导致整体的熵偏大。

（三）样本的其他特征

1. 频率最高的字符

各样本出现频率最高的五个字符，虽然不同样本间有差别，但总体上出现频率最高的几个字符都相同。

来源	出现频率最高的五个字符
维基百科(英文)	空格, e, t, a, i
ABC News & CNN	空格, e, t, a, i
英文小说	空格, e, t, a, o
维基百科(中文)	的, 在, 国, 为, 是
人民网	的, 人, 中, 一, 国
纵横中文网	的, 一, 了, 是, 不

表3

2. 汉字、词的频率和特征

汇集所有中文样本，一共出现了 7474 个汉字，出现次数大于 1 的有 6833 个，出现频率最高的五个字如下，都是虚词，“的”是使用频率最高的汉字。

汉字	频率	概率
的	1645327	3.05%
一	648783	1.20%
是	541520	1.00%
了	471559	0.88%
在	470710	0.87%

表4

中文样本中，除去所有虚词后，频率最高的十个字、词及其频率如下：

字	频率	概率	实词	频率	概率
他	167360	0.30%	发展	72337	0.13%
人	105370	0.19%	一个	65743	0.12%
我	104628	0.20%	没有	57517	0.11%
年	95612	0.17%	中国	50401	0.09%
你	95164	0.18%	自己	48153	0.09%
中	94352	0.18%	他们	48089	0.09%
不	90655	0.17%	工作	46828	0.09%
上	84528	0.16%	我们	43189	0.08%
被	78960	0.15%	已经	37475	0.07%
对	78061	0.15%	国家	37285	0.07%

表5

汉字中出现频率最高的字是“他”，可能的原因是，数据来源是百科、新闻和小说，大多都需要使用客观第三人称来叙述。

3. 英文字母、单词的频率和特征

汇集所有英文样本，英文出现频率最高的五个字母：

字母	频率	概率
空格	51926050	17.06%
e	30907740	10.16%
t	22531193	7.40%
a	20943067	6.88%
i	18763347	6.17%

表6

英文样本中，除去所有虚词后，频率最高的十个单词及其频率如下：

单词	频率	概率
said	251357	0.29%
like	171906	0.20%
time	138165	0.16%
just	135769	0.16%
know	116123	0.14%
did, didn't, don't	98443, 97871, 94073	0.11%, 0.11%, 0.11%
people	87308	0.10%
way	86606	0.10%
new	84116	0.10%
going	80239	0.09%

表7

英文同理，*said* 是出现频率最高的单词，原因也很可能是这些文本来源都需要从客观第三人称来叙述。

4. 验证齐夫定律

将中文和英文样本分别汇集之后，计算频率最高的 10 个字（字母）的频率和排名的对数，并将其绘制成图像，如下所示：

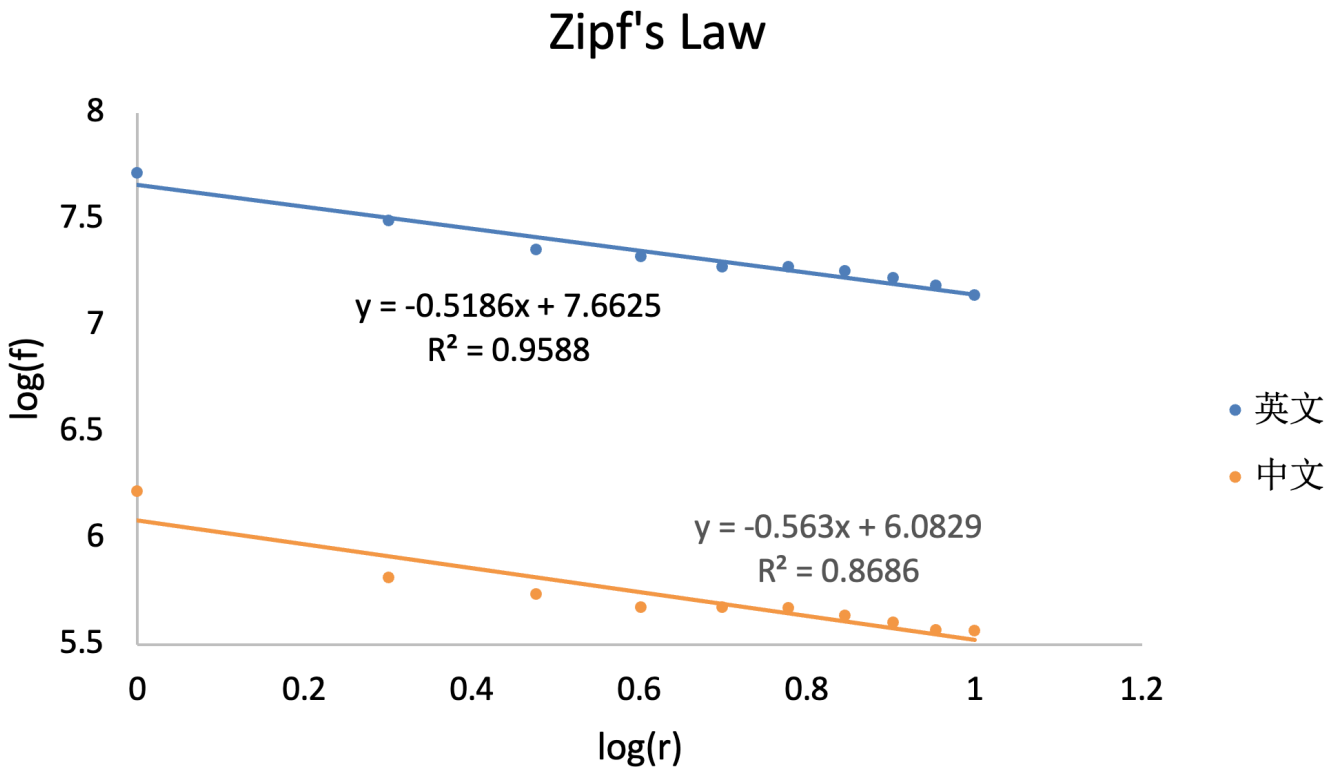

图 4-中英文样本频率和排序位次的对数关系

$log(r)$ 与 $log(f)$ 的取值关系近似为一条直线，基本符合齐夫定律。

五、不足

从上面的数据可以看到，本次选取的数据大多倾向于从客观第三人称来叙述，数据的覆盖面可能不够广，样本不够丰富。并且，相较于英文样本，本次选取的中文样本量较少，可能对结论的准确性和普适性有负面的影响。