计算语言学

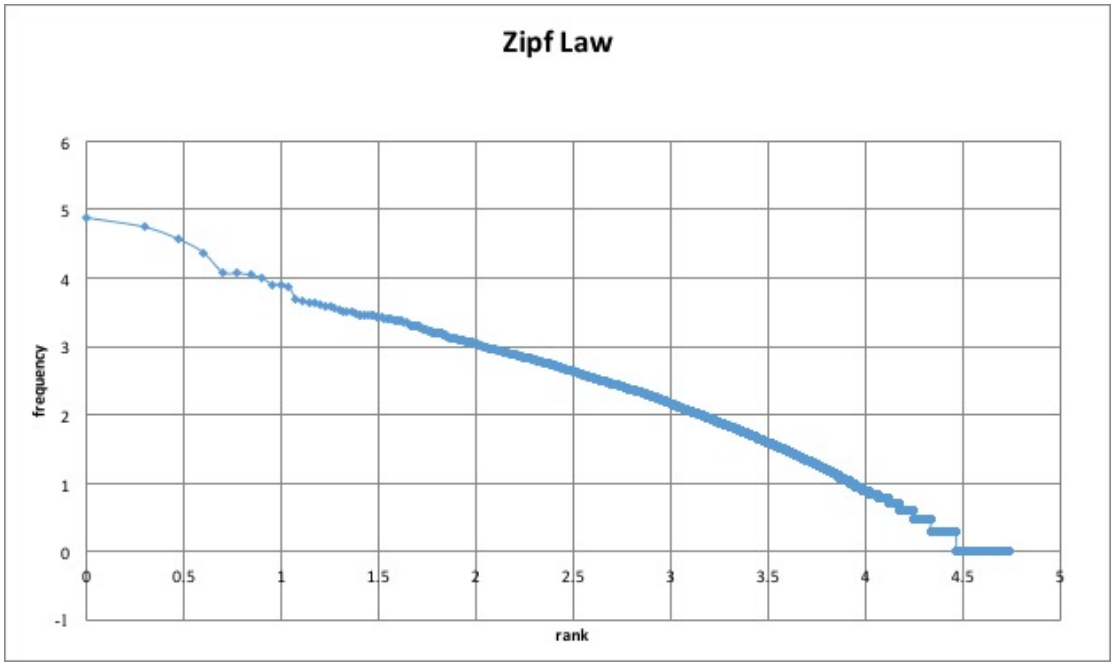
2017211382 陈璐 交叉研 17

1. **Zipf 定律曲线生成**

1) 统计已人工分词的语料库中所有词的**词频**，并按词频由高到低进行排序，生成文件“1.txt”。生成的统计结果前 20 个如下图。

```
1  ，	74921
2  的	54476
3  。	35983
4  、	23116
5  在	12022
6  了	11557
7  和	10919
8  是	9819
9  “	7970
10 ”	7943
11 一	7335
12 为	4747
13 有	4641
14 ）	4317
15 （	4317
16 不	4095
17 对	3800
18 上	3795
19 中	3699
20 中国	3358
```

2) 基于该词频表，利用 Excel 生成 Zipf 定律的曲线图，生成文件“2.jpeg”。如下图所示。



利用 Excel 分别求出 log(rank)和 log(frequency)，在对二者做散点图，得出上图。可以从图中曲线看出，曲线呈现极大的规律性，斜率逼近-1，几乎是一条 45 度角的直线。这是因为 rank * frequency 接近一个常数。这说明一个词的出现次数跟它的等级序号成反比。

在 EXCEL 中算出 rank * frequency 如下图

38	工作	2412	91656
39	将	2410	93990
40	地	2374	94960
41	以	2362	96842
42	企业	2329	97818
43	新	2245	96535
44	大	2244	98736
45	记者	2143	96435
46	国家	2046	94116
47	从	2033	95551
48	我们	2027	97296
49	两	1952	95648
50	》	1940	97000
51	《	1940	98940
52	都	1908	99216
53	一个	1889	100117
54	我	1802	97308
55	1月	1781	97955
56	建设	1744	97664
57	问题	1712	97584
58	着	1683	97614
59	来	1620	95580
60	市场	1614	96840
61	已	1609	98149
62	全国	1593	98766
63	人民	1579	99477
64	并	1573	100672
65	把	1563	101595
66	还	1553	102498

上图第一列是 rank，第二列是词，第三列是 frequency，第四列是 rank 和 frequency 的乘积。rank 和 frequency 的乘积接近 C= 100000. 经过统计得到整个《人民日报》的语料单词个数为 N = 1120721，N / 10 约等于 C

2. 基于 n-gram 的句子概率计算

分别使用 **unigram** 和 **bigram** 计算以下两个句子的概率

- 扶贫 开发 工作 取得 很 大 成绩 （句子 1）
- 扶贫 开发 工作 得到 很 大 成绩 （句子 2）
- <BOS> 扶贫 开发 工作 取得 很 大 成绩 （<句子 1a）
- <BOS> 扶贫 开发 工作 得到 很 大 成绩 （<句子 2a）
- <BOS> 扶贫 开发 工作 取得 很 大 成绩 <EOS> （句子 1b）
- <BOS> 扶贫 开发 工作 得到 很 大 成绩 <EOS> （句子 2b）

生成文件“3.txt”，格式如下：

- 第一行输出句子 1 的 unigram 句子概率，并分别输出每个 unigram 的概率。
- 第二行输出句子 1 的 bigram 句子概率，并分别输出每个 bigram 的条件概率。
- 第三行输出句子 1a 的 bigram 句子概率，并补充输出与<BOS>相关的 bigram 条件概率。

- 第四行输出句子 1b 的 bigram 句子概率，并补充输出与<BOS>和<EOS>相关的 bigram 条件概率。

- 对句子 2 同上依次处理。

- 行内以 tab 分隔，概率输出取 log（10 为底）结果，小数点后保留 6 位。第一列是句子概率，后面是单词的 unigram 或者 bigram 的概率

```
-22.281685	-3.489591	-3.364652	-2.667120	-3.284575	-3.089503	-2.698475	-3.687770
-14.012473	-0.991705	-2.207724	-1.919979	-2.463893	-0.734686	-2.204895
-14.113166	-3.590284	-0.991705	-2.207724	-1.919979	-2.463893	-0.734686	-2.204895
-14.793653	-3.590284	-0.991705	-2.207724	-1.919979	-2.463893	-0.734686	-2.204895	-0.680487
-22.474899	-3.489591	-3.364652	-2.667120	-3.477789	-3.089503	-2.698475	-3.687770
-14.406595	-0.991705	-2.207724	-2.683407	-2.094588	-0.734686	-2.204895
-14.507289	-3.590284	-0.991705	-2.207724	-2.683407	-2.094588	-0.734686	-2.204895
-15.187775	-3.590284	-0.991705	-2.207724	-2.683407	-2.094588	-0.734686	-2.204895	-0.680487
```

1) 两句子之间概率进行比较

   i. $P_{unigram}(S_1) > P_{unigram}(S_2)$

   ii. $P_{bigram}(S_1) > P_{bigram}(S_2)$

   iii. $P_{bigram}(S_{1a}) > P_{bigram}(S_{2a})$

   iv. $P_{bigram}(S_{1b}) > P_{bigram}(S_{2b})$

   可以看出句子 1 比句子 2 是更常用。

   $\log[P_{s1,bigram}(取得|工作)] = -1.919979$

   $\log[P_{s2,bigram}(得到|工作)] = -2.667120$

   两个句子的不同之处在于“工作取得”和“工作得到”，由上面的 log 概率可以看出“工作取得”更加常用。在日常用语中也的确是“工作取得”使用频率更高

2) 两句子内 bigram 和 unigram 概率进行比较

   $\log[P_{bigram}(S_1) / P_{unigram}(S_1)] = \log[P_{bigram}(S_1)] - \log[P_{unigram}(S_1)] \approx 8.27$

   $\log[P_{bigram}(S_2) / P_{unigram}(S_2)] = \log[P_{bigram}(S_2)] - \log[P_{unigram}(S_2)] \approx 8.07$

   bigram 比 unigram 精确很多。bigram model 比 unigram model 可以更有效地刻画一句话

3) 两句子内 bigram 之间概率进行比较

   对于 $P_{bigram}(S_1)$、$P_{bigram}(S_{1a})$、$P_{bigram}(S_{1b})$的乘项逐步递增。句子 1a 相较于句子 1 将句首考虑进来，句子 1b 相较于句子 1a 进一步考虑了句尾。由于增加了乘项，算出来的句子概率相应变得小一些，句子 bigram 概率还在同一量级内，但是这样计算句子得能更加精准。可以说考虑的元素越多，对句子的刻画会更佳精确，对句子的刻画能力更强。

4) unigram 和 bigram 的空间大小进行比较

   Unigram 空间大小:55411

   Bigram 空间大小:457709

   Bigram 空间大小几乎是 unigram 空间大小的平方，

5) 下面给出我对各个概率的计算方式

i. **对句子 1 的 unigram 句子概率**

$$P_{unigram}(s1) = P(扶贫) * P(开发) * P(工作) * P(取得) * P(很) * P(大) * P(成绩)$$

ii. **对句子 1 的 bigram 句子概率**

$$P_{bigram}(s1) = P(扶贫) * P(开发|扶贫) * P(工作|开发) * P(取得|工作) * P(很|取得) * P(大|很) * P(成绩|大)$$

iii. **对句子 1a 的 bigram 句子概率**

$$P_{bigram}(s1a) = P(扶贫| < BOS >) * P(开发|扶贫) * P(工作|开发) * P(取得|工作) * P(很|取得) * P(大|很) * P(成绩|大)$$

iv. **对句子 1b 的 bigram 句子概率**

$$P_{bigram}(s1b) = P(扶贫| < BOS >) * P(开发|扶贫) * P(工作|开发) * P(取得|工作) * P(很|取得) * P(大|很) * P(成绩|大) * P(< EOS > |成绩)$$

句子 2 概率计算方式相仿