文本挖掘 class1

kcao

3/7

1.Shell

[kcao@h1-lgl test]$ sh fun.sh *The_little_Prince.txt* *100*

$1 $2

```
#!/bin/bash
if [ ! -e $1 ] || [ -z $2 ];then
    echo "! ! ! 请分别输入文件和行数"
    exit 1
elif [ -e $1.output ];then
    rm -rf $1.output
fi

cat $1 |tr -sc [:alnum:] "\n"|tr [:upper:] [:lower:] >$1.a.pure.txt
total=`cat $1.a.pure.txt|wc -l `

for i in `seq 1 $[$total /$2]`            #循环次数
do
    head -n $[ i*$2 ] $1.a.pure.txt>tmp_file
    words=`cat tmp_file|wc -l`
    token=`sort tmp_file|uniq |wc -l`
    ratio=` echo "scale=4;$token / $words "|bc`
    echo -e "$[ i*$2 ]\t$ratio\t$token" >>$1.output
done
```

①判断$1,$2时候输入，及其$1.output是否存在

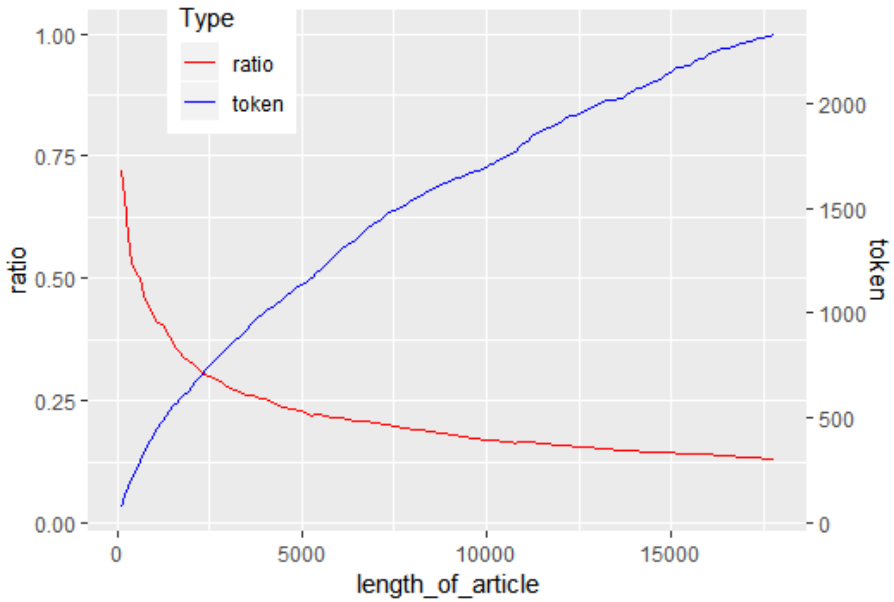
②循环次数为total/$2

③将每次ratio,token保存在$1.output

```
[kcao@h1-lgl test]$ echo $[ 8/3 ]
2
[kcao@h1-lgl test]$ echo $[ 2/3 ]
0
[kcao@h1-lgl test]$ echo "scale=4;2/3 "|bc
.6666
[kcao@h1-lgl test]$ awk 'BEGIN{printf "%.4f\n",2/3}'
0.6667
```

2.R

```
Article<-read.table("./The_little_Prince.txt.output",header = F)
colnames(Article)<-c("length_of_article","ratio","token")
p<-(ggplot(Article,aes(x=length_of_article))
  +geom_line(aes(y=ratio,col="ratio"))
  +geom_line(aes(y=token/2332,col = "token"))
  +scale_y_continuous(sec.axis = sec_axis(~.*2332, name =  "token"))
  +scale_color_manual(values = c("red","blue"))
  +theme(legend.position = c(0.2,0.9))
  +labs(color="Type")
)
p
```



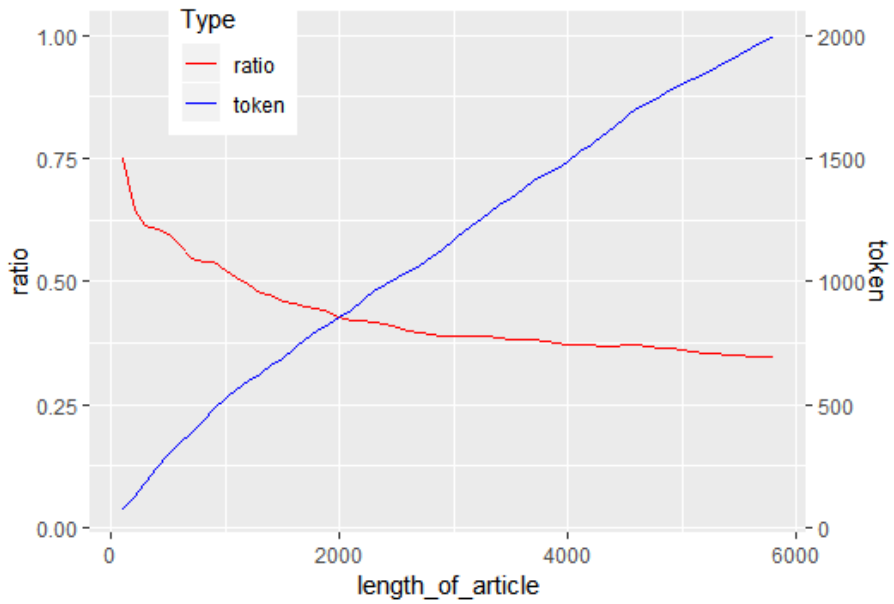
《The_little_Prince.txt》

The_little_Prince.txt [2].output

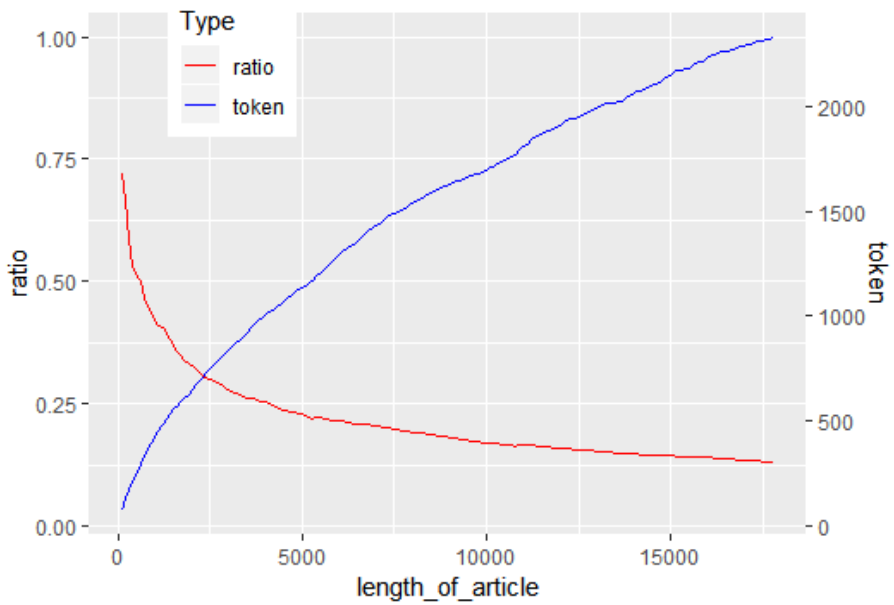
156	15600	.1405	2193
157	15700	.1402	2202
158	15800	.1398	2210
159	15900	.1394	2217
160	16000	.1395	2233
161	16100	.1393	2244
162	16200	.1388	2250
163	16300	.1384	2256
164	16400	.1378	2261
165	16500	.1372	2264
166	16600	.1363	2264
167	16700	.1358	2268
168	16800	.1354	2276
169	16900	.1350	2283
170	17000	.1345	2287
171	17100	.1340	2292
172	17200	.1336	2299
173	17300	.1330	2302
174	17400	.1326	2308
175	17500	.1321	2313
176	17600	.1316	2317
177	17700	.1311	2322
178	17800	.1310	2332

tips:每次画图需要修改“2332”此处的值

3.比较



CET4



《The_little_Prince.txt》

CET4 6000，*《小王子》* 13000

Token=2000

小王子文本复杂度相对较低