NewDictStructure

YANG YANG yy2819

2018-11-28

```
library(stringr)
url ="http://www.gutenberg.org/cache/epub/1260/pg1260.txt"
url <- "http://www.gutenberg.org/cache/epub/55/pg55.txt"
raw <- readLines(url, encoding = "UTF-8")
vec <- unlist( str_split(raw," ") )
vocab=unique(vec)
vocab=vocab[-which(vocab=="")]

n=length(vocab)
p_vocab=1:n
names(p_vocab)=vocab
v1=vocab[1]
vn=vocab[n]
vm=vocab[n%/%2]

#check
p_vocab[v1]
```

```
## The
##   1
```

```
p_vocab[vn]
```

```
## newsletter
##       6071
```

```
# 10k calls
system.time(for (i in 1:10000) temp=1)
```

```
##    user  system elapsed
##    0.01    0.00    0.01
```

```
system.time(for (i in 1:10000) temp=p_vocab[v1])
```

```
##    user  system elapsed
##    0.27    0.19    0.57
```

```
system.time(for (i in 1:10000) temp=p_vocab[vm])
```

```
##    user  system elapsed
##    0.42    0.26    0.79
```

```
system.time(for (i in 1:10000) temp=p_vocab[vn])
```

```
##    user  system elapsed
##    0.64    0.19    0.83
```

We see that the time cost of vector calls linearly increases with index.

what if p_vocab is a matrix

```r
matrix_vocab=matrix(rep(p_vocab,20),ncol = 20)
rownames(matrix_vocab)=vocab
matrix_vocab[vn,1]
```

```
## newsletter
##       6071
```

```r
system.time(for (i in 1:10000) temp=matrix_vocab[vn,1])
```

```
##    user  system elapsed
##    0.64    0.27    0.97
```

```r
my_dict <- new.env()
for(i in 1:n){
  my_dict[[vocab[i]]]=i
}
#check
my_dict[[v1]]
```

```
## [1] 1
```

```r
my_dict[[vn]]
```

```
## [1] 6071
```

```r
# 10k calls
system.time(for (i in 1:10000) temp=my_dict[[v1]])
```

```
##    user  system elapsed
##       0       0       0
```

```r
system.time(for (i in 1:10000) temp=my_dict[[vn]])
```

```
##    user  system elapsed
##       0       0       0
```

```r
# 10m calls
system.time(for (i in 1:10000000) temp=my_dict[[v1]])
```

```
##    user  system elapsed
##    1.92    0.00    1.95
```

```r
system.time(for (i in 1:10000000) temp=my_dict[[vn]])
```

```
##    user  system elapsed
##    1.69    0.00    1.73
```

We see that the time cost is almost independent of index, and much faster.(1000 times faster on average while the corpus has 28020 unique words)