NewDictStructure

YANG YANG yy2819

2018-11-28

```
library(stringr)
url <- "http://www.gutenberg.org/cache/epub/1260/pg1260.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
##      28020
```

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

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

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

```
##    user  system elapsed
##    0.79    0.77    1.60
```

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

```
##    user  system elapsed
##    2.27    0.68    2.95
```

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

```
##    user  system elapsed
##    4.07    0.78    4.93
```

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

what if p_vocab is a matrix

```
matrix_vocab=matrix(rep(p_vocab,20),ncol = 20)
rownames(matrix_vocab)=vocab
```

```
matrix_vocab[vn,1]
```

```
## newsletter
##      28020
```

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

```
##    user  system elapsed
##    4.34    0.95    5.58
```

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

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

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

```
## [1] 28020
```

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

```
##    user  system elapsed
##    0.01    0.00    0.02
```

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

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

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

```
##    user  system elapsed
##    2.01    0.00    2.12
```

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

```
##    user  system elapsed
##    2.17    0.00    2.17
```

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)