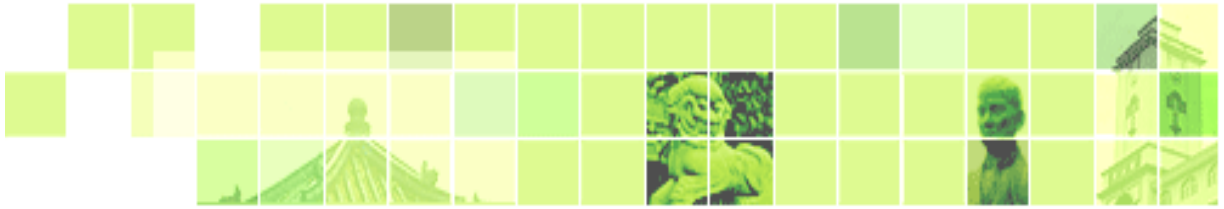
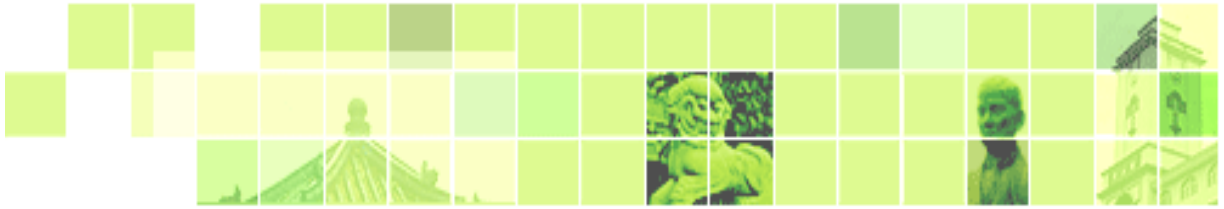
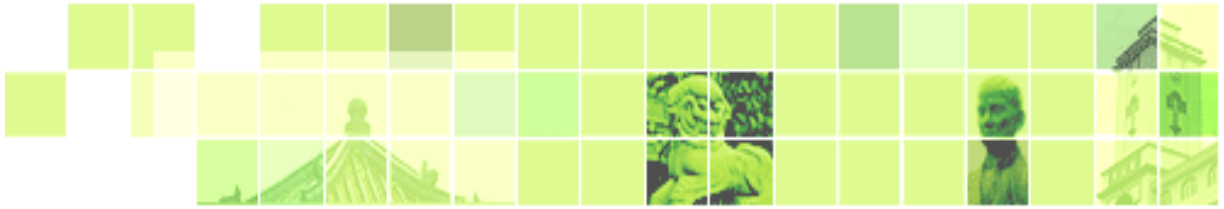


第1节练习：使用Unix/Linux工具进行文本处理



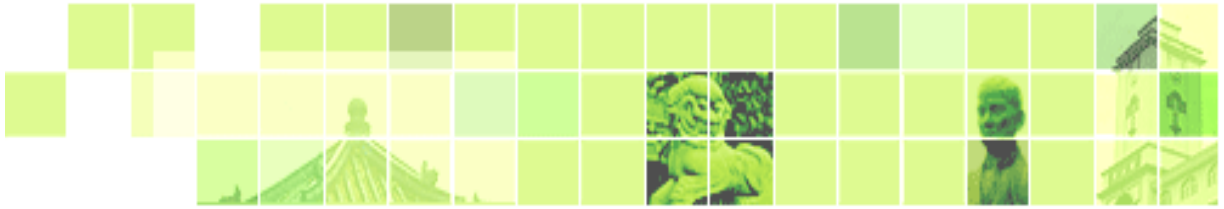
Unix for Poets

- 文本无处不在
  - 网络
  - 字典，语料库，电子邮件
  - 数十亿词汇
- 我们能做些什么呢？
- 可以利用Unix的命令行做一些简单的操作
- 有时甚至比编写python程序快得多



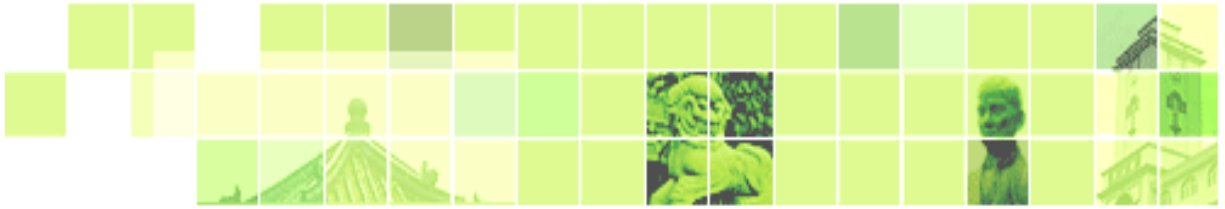
将要进行的练习

1. 统计文本中的单词数
2. 以不同方式对单词列表进行排序
   - Ascii order
   - “rhyming”order
3. 抽取字典中的有用信息
4. 计算N-Gram（语言模型）统计数据
5. 处理标记文本中的词性



工具

- **grep**: search for a pattern (regular expression)
- **sort**
- **uniq –c** (count duplicates)
- **tr** (translate characters)
- **wc** (word – or line – count)
- **sed** (edit string -- replacement)
- **cat** (send file(s) in stream)
- **echo** (send text in stream)
- **cut** (columns in tab-separated files)
- **paste** (paste columns)
- **head**
- **tail**
- **rev** (reverse lines)
- **comm**
- **join**
- **shuf** (shuffle lines of text)



先决条件：获取文本文件

- myth: ssh into a myth and then do:

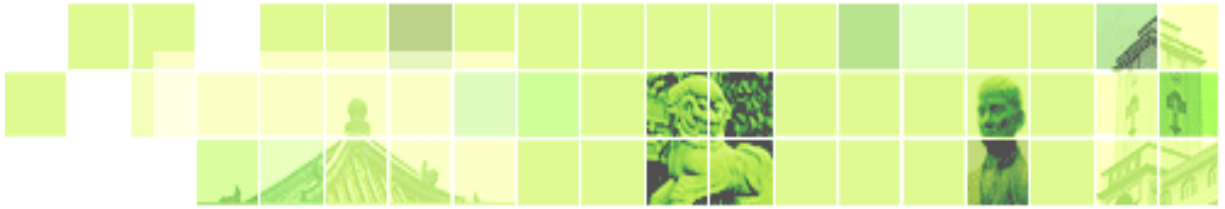
```
scp cardinal: /afs/ir/class/cs124/nyt_200811.txt.gz .
```

- Or if you're using your own Mac or Unix laptop, do that or you could download, if you haven't already:

http://cs124.stanford.edu/nyt_200811.txt.gz

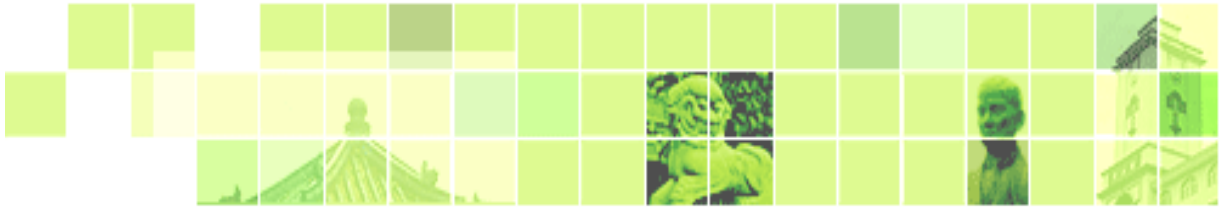
- Then

```
gunzip nyt_200811.txt.gz
```



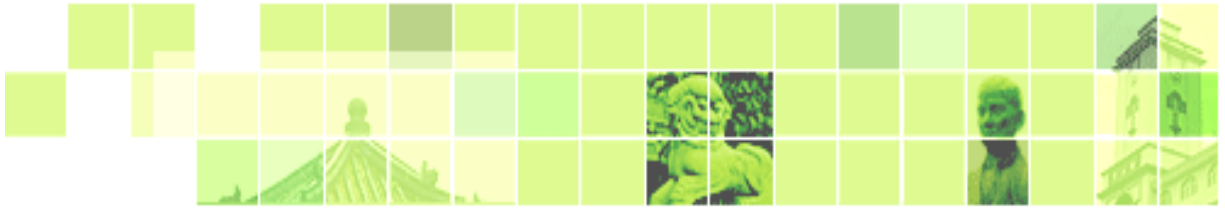
先决条件

- Unix中的“man”命令
  - e.g., **`man tr`** (shows command options; not friendly)
- 输入/输出重定向：
  - **`>`** **“ouput to a file”**
  - **`<`** **“input from a file”**
  - **`|`** **“pipe”（组合）**
- **CTRL-C**



练习1：统计文本中的单词数

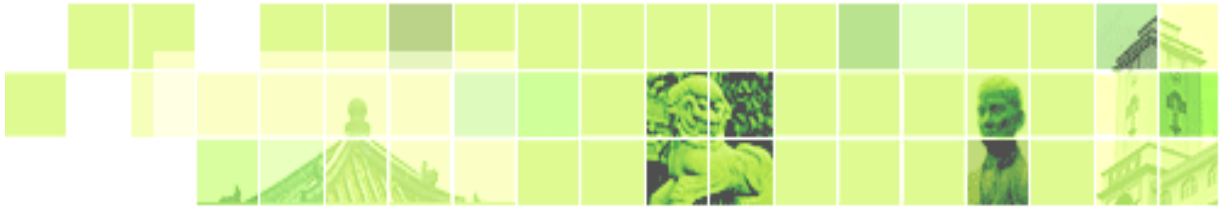
- Input: text file (nyt_201811.txt) (after it's gunzipped)
- Output: 文件中的单词表及其对应频次
- Algorithm
  1. Tokenize (tr)
  2. Sort (sort)
  3. Count duplicates (uniq -c)
- 阅读手册页并找出如何将它们组合在一起



解答

- `tr -sc 'A-Za-z' '\n' < nyt_200811.txt | sort | uniq -c`

```
25476 a
 1271 A
    3 AA
    3 AAA
    1 Aalborg
    1 Aaliyah
    1 Aalto
    2 aardvark
```

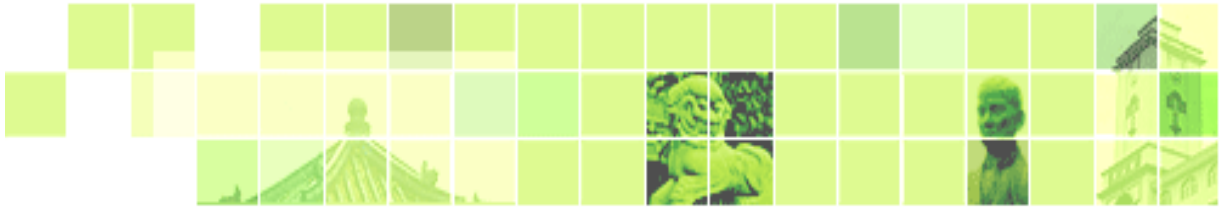


一些输出

- **tr -sc 'A-Za-z' '\n' < nyt_200811.txt | sort | uniq -c | head -n 5**

```
25476 a
 1271 A
    3 AA
    3 AAA
    1 Aalborg
```

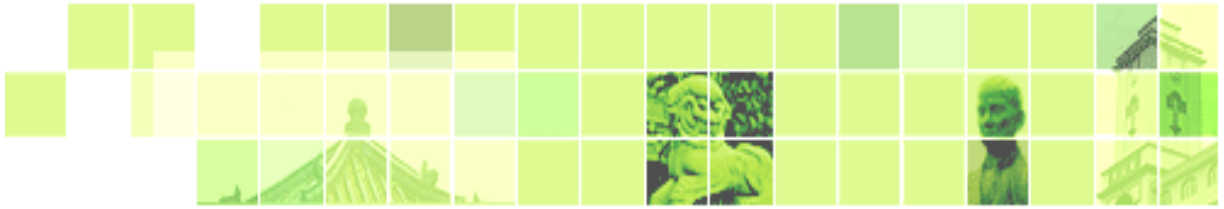
- **Tr -sc 'A-Za-z''\n' < nyt_201811.txt | sort | uniq -c | head**
- 列出前10行结果
- 可以省略“-n”但不建议这样做



扩展计数练习

1. 通过将所有大写字母转换成小写字母来合并大小写
   - Hint: Put in a second tr command

2. 不同的元音序列的频次
   - Hint: Put in a second tr command

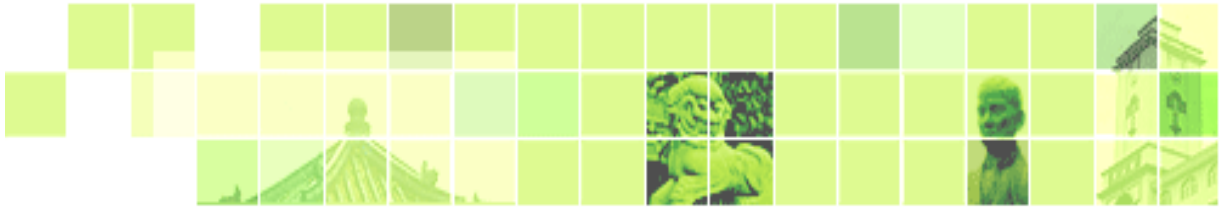


解答

1. 通过将所有大写字母转换成小写字母来合并大小写

```
tr -sc 'A-Za-z' '\n' < nyt_200811.txt | tr 'A-Z' 'a-z'
| sort | uniq -c
or
tr -sc 'A-Za-z' '\n' < nyt_200811.txt | tr '[:upper:]'
'[:lower:]' | sort | uniq -c
```

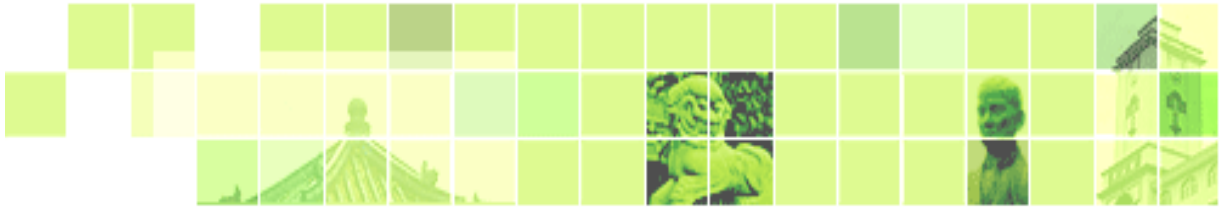
- 用换行符分割，把连续的字符以单独的字符表示
- 将所有的大写字母替换成小写字母
- 以字母序排列
- 合并重复项并计数



解答

2. 不同的元音序列的频次 (e.g., ieu)

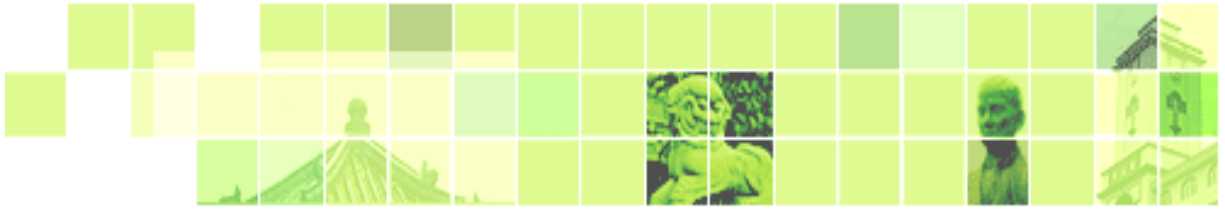
```
tr -sc 'A-Za-z' '\n' < nyt_200811.txt | tr
'A-Z' 'a-z' | tr -sc 'aeiou' '\n' | sort |
uniq -c
```



排序与反转（Sort and reversing lines of the text）

- `sort`
- `sort -f` 忽略大小写
- `sort -n` 数字顺序
- `sort -r` 倒序
- `sort -nr` 数字倒序

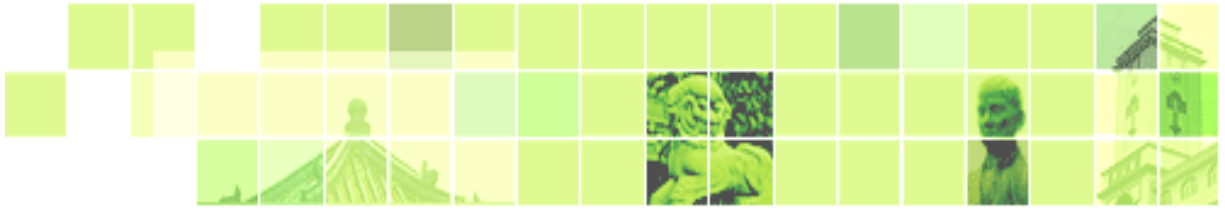
- `echo “Hello” | rev`



计数与排序练习

1. 找出 NYT 中最常见的50个单词
   - Hint: Use sort a second time, then head

2. 找出 NYT 中以“zz”为结尾的单词
   - Hint: Look at the end of a list of reversed words
   - tr ‘A-Z’ ‘a-z’ < filename | tr –sc ‘A-Za-z’ ‘\n’ | rev | sort | rev | uniq -c



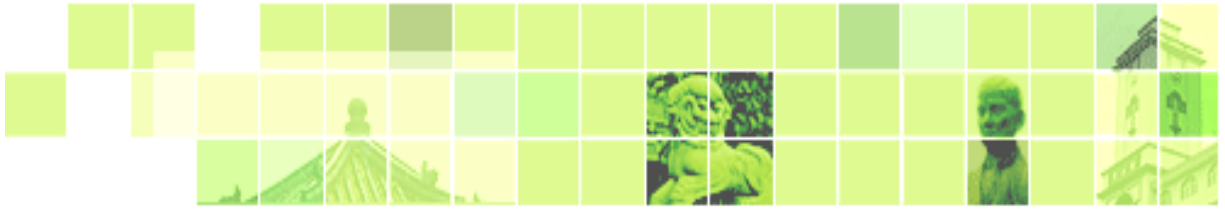
计数与排序练习

1. 找出 NYT 中最常见的50个单词

```
tr -sc 'A-Za-z' '\n' < nyt_200811.txt |
sort | uniq -c | sort -nr | head -n 50
```

2. 找出 NYT 中以“zz”为结尾的单词

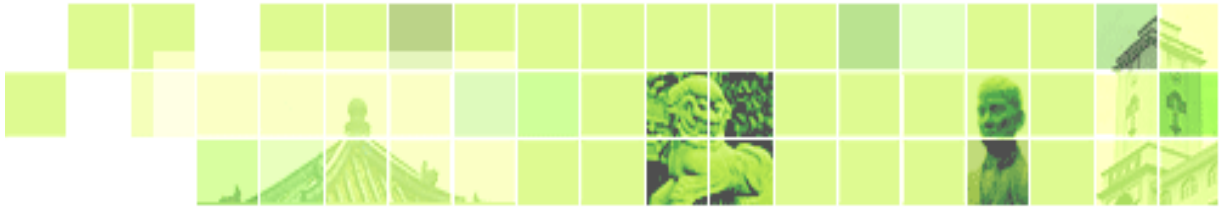
```
tr -sc 'A-Za-z' '\n' < nyt_200811.txt | tr
'A-Z' 'a-z' | rev | sort | uniq -c | rev |
tail -n 10
```



N-gram

N-gram的基本思想是将文本内容按字节流进行大小为N的滑动窗口操作，形成长度为N的字节片段序列，每个字节片段即为gram，对全部gram的出现频度进行统计，并按照设定的阈值进行过滤，形成keygram列表，即为该文本的特征向量空间。

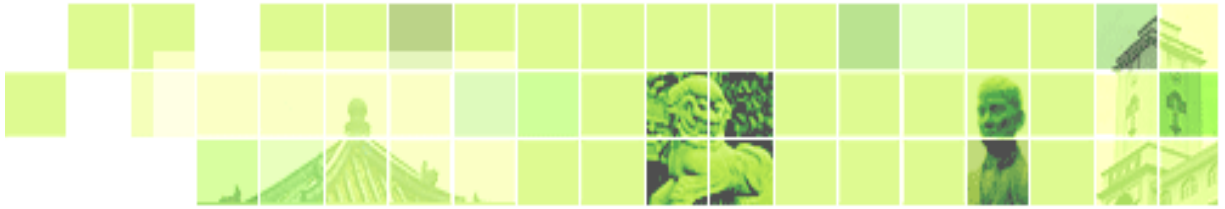
Bigram为二元语法，此时N=2



Bigrams = word pairs and their counts

算法：

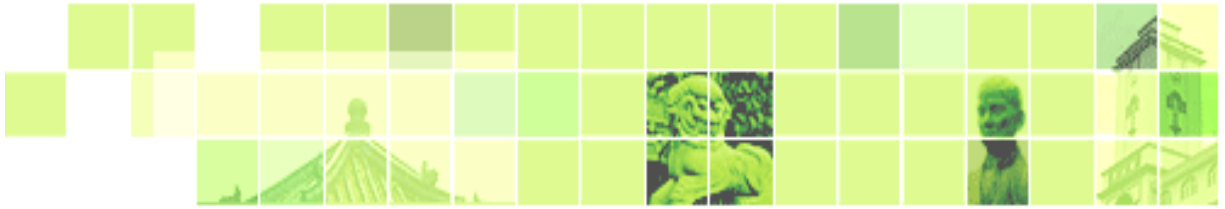
1. Tokenize by word
2. Create two almost-duplicate files of words, off by one line, using **tail**
3. **Paste** them together so as to get $word_i$ and $word_{i+1}$ on the same line
4. Count



Bigrams

- `tr -sc 'A-Za-z' '\n' < nyt_200811.txt > nyt.words`
- `tail -n +2 nyt.words > nyt.nextwords`
- `paste nyt.words nyt.nextwords > nyt.bigrams`
- `head –n 5 nyt.bigrams`

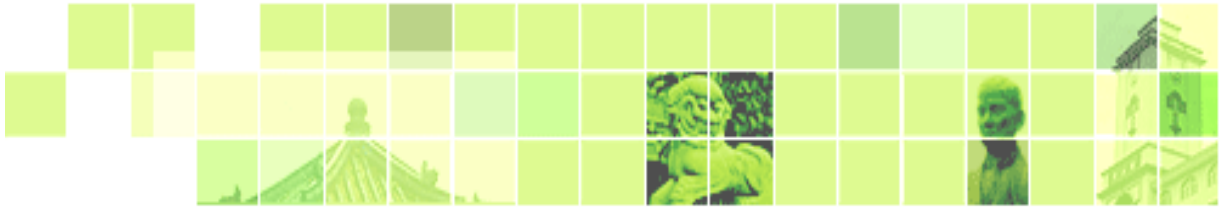
```
KBR       said
said      Friday
Friday    the
the       global
global    economic
```



练习

1. 找出10个最常见的二元语法

2. 找出10个最常见的三元语法



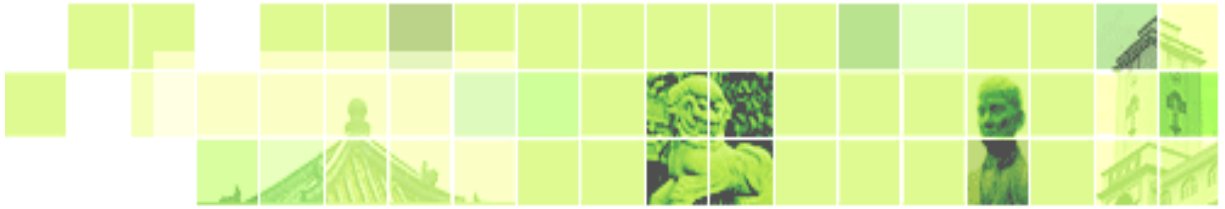
解答

1. 找出10个最常见的二元语法

```
tr 'A-Z' 'a-z' < nyt.bigrams | sort | uniq
-c | sort -nr | head -n 10
```

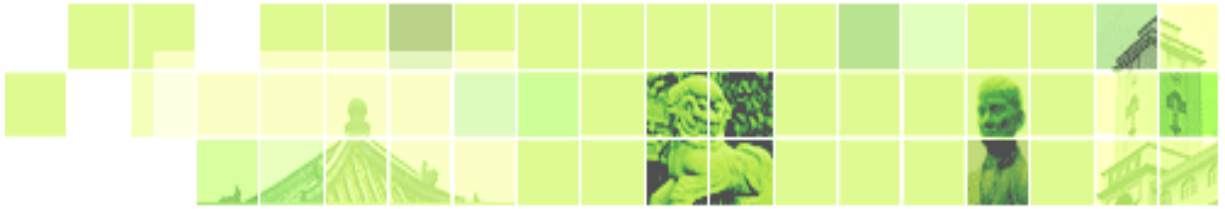
1. 找出10个最常见的三元语法

```
tail -n +3 nyt.words > nyt.thirdwords
paste nyt.words nyt.nextwords nyt.thirdwords >
nyt.trigrams
cat nyt.trigrams | tr "[:upper:]" "[:lower:]" | sort |
uniq -c | sort -rn | head -n 10
```



Grep

- Grep算法查找指定为正则表达式的模式
  - globally search for regular expression and print
- 查找以 –ing 为结尾的词
  - `grep 'ing$' nyt.words | sort | uniq -c`
- `grep gh` 保留含有“gh”的行
- `grep '^con'` 保留以“con”为开头的行
- `grep 'ing$'` 保留以“ing”为结束的行
- `grep -v gh` 保留不含“gh”的行
- `egrep` 扩展语法



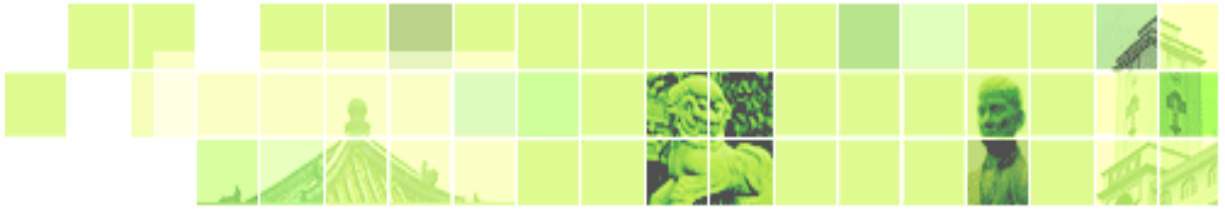
行，单词，字符计数

- **wc nyt_200811.txt**

```
140000 1007597 6070784 nyt_200811.txt
```

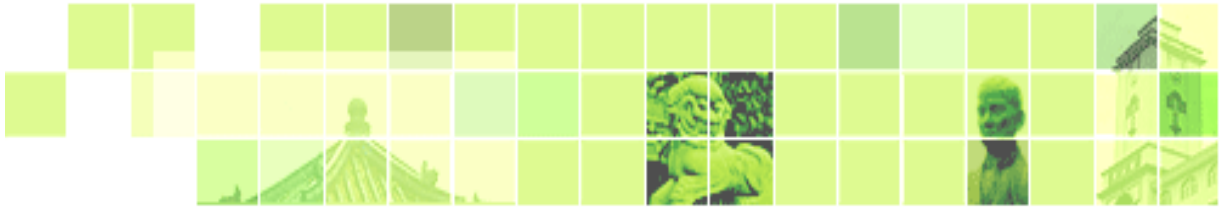
- **wc -l nyt.words**

```
1017618 nyt.words
```



练习：grep & wc

- NYT文件中，有多少字母全为大写的单词？
- 共有多少4位字母的单词？
- 共有多少不含元音的单词？
  - 他们属于哪种类型？
- 共有多少单音节单词？
  - “1 syllable” means that the ones with exactly one vowel（只含一个元音）



练习：grep & wc

- NYT文件中，有多少字母全为大写的单词？

```
grep -P '^[A-Z]+$' nyt.words | wc
```

- 共有多少4位字母的单词？

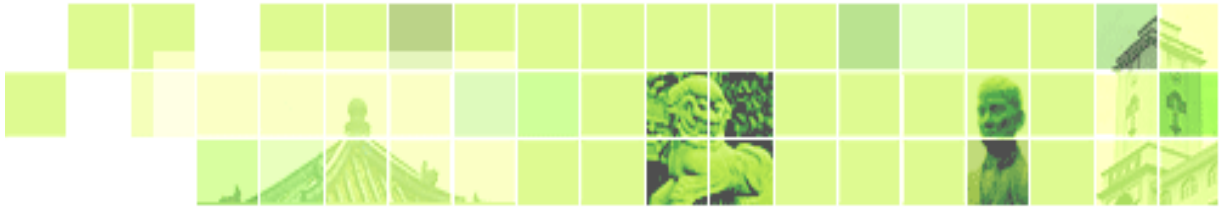
```
grep -P '^[a-zA-Z]{4}$' nyt.words | wc
```

- 共有多少不含元音的单词？
  - 他们属于哪种类型？

```
grep -v '[AEIOUaeiou]' nyt.words | sort | uniq | wc
```

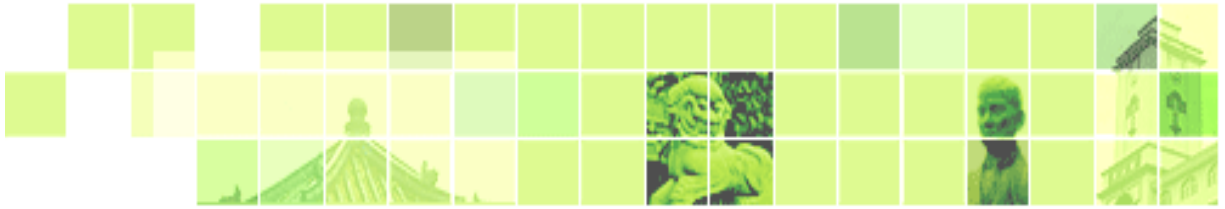
- 共有多少单音节单词？

```
tr 'A-Z' 'a-z' < nyt.words | grep -P
'^[^aeiouAEIOU]*[aeiouAEIOU]+[^aeiouAEIOU]*$' | uniq | wc
```



Sed

- 需要对文件中的字符串进行系统地更改时，使用sed命令
- 基于行：可以选择指定行（通过regex或行号），并制定一个即将进行的regex替换
- 例：将所有的“George”替换为“Jane”
- **`sed 's/George/Jane/' nyt_200811.txt | less`**



Cut – 制表符分隔的文件

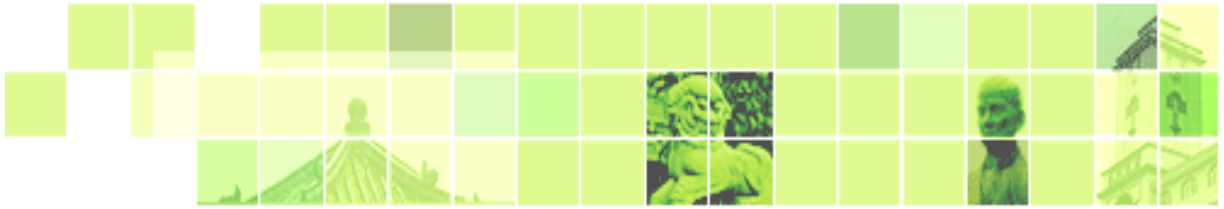
- Frequency of different parts of speech

```
cut -f 4 parses.conll | sort | uniq -c | sort
-nr
```

- Get just words and their parts of speech

```
cut -f 2,4 parses.conll
```

- 可以用“cut -d”处理逗号分割的文件



Thank you !