數據分析師假日精修班
Lab5

David Chiu
2016/07/30

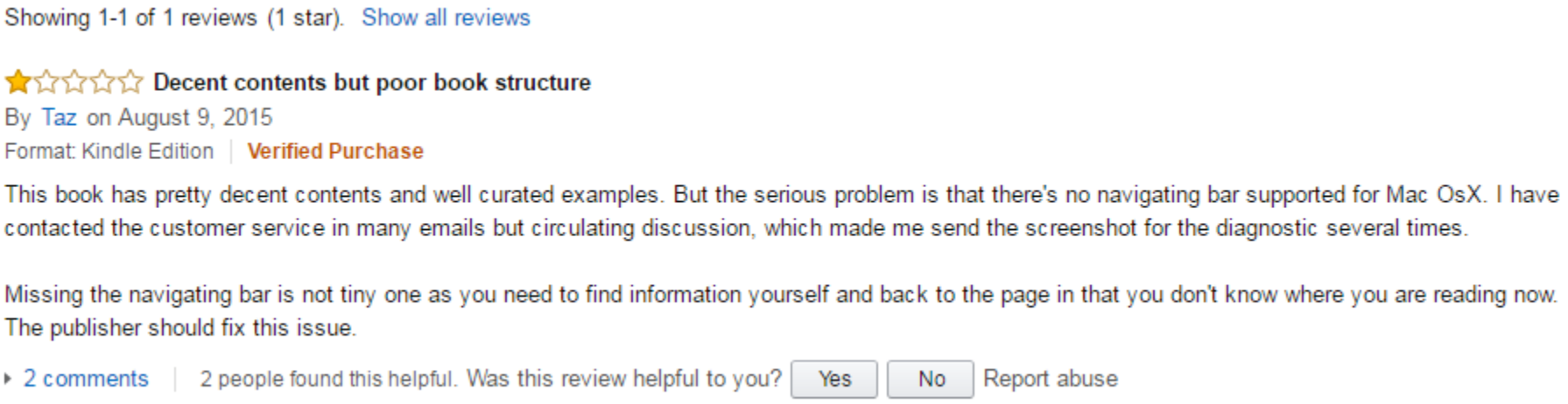
文字探勘的重要性

星星代表的意思?


amazon
Try Prime
Books
NEW & INTERESTING FINDS FROM ACROSS AMAZON
Departments
Browsing History
David's Amazon.com
Today's Deals
Gift Cards & Registry
Sell
Help
Hello, David
Your Account
Try Prime
Lists
Cart
Books
Advanced Search
New Releases
Best Sellers
The New York Times Best Sellers
Children's Books
Textbooks
Textbook Rentals
Sell Us Your Books
Best Books of the Month
Books › Computers & Technology › Databases & Big Data
Machine Learning with R Cookbook
Flip to back
Machine Learning With R Cookbook - 110 Recipes for Building Powerful Predictive Models with R Paperback – March 26, 2015
by Chiu (David Chiu) Yu-Wei (Author)
10 customer reviews
See all 2 formats and editions
Kindle
$31.99
Paperback
$39.99
Read with Our Free App
9 Used from $43.64
16 New from $39.99
Key Features
Apply R to simplify predictive modeling with short and simple code
Use machine learning to solve problems ranging from small to big data
Build a training and testing dataset from the churn dataset, applying different classification methods
Share
<Embed>
Buy New
$39.99
Qty: 1
In Stock.
Ships from and sold by Amazon.com.
Gift-wrap available.
Add to Cart
This item ships to Taipei, Taiwan; Republic of China. Learn more
Pay in TWD with 1-Click
Price in TWD: 1,363.80
Change 1-Click payment to USD
Ship to:
David Chiu- Taipei

內部的評論反應使用者的真正感覺

- But the serious problem is that there's no navigating bar supported for Mac OsX



Showing 1-1 of 1 reviews (1 star). Show all reviews

★☆☆☆☆ **Decent contents but poor book structure**

By Taz on August 9, 2015

Format: Kindle Edition | **Verified Purchase**

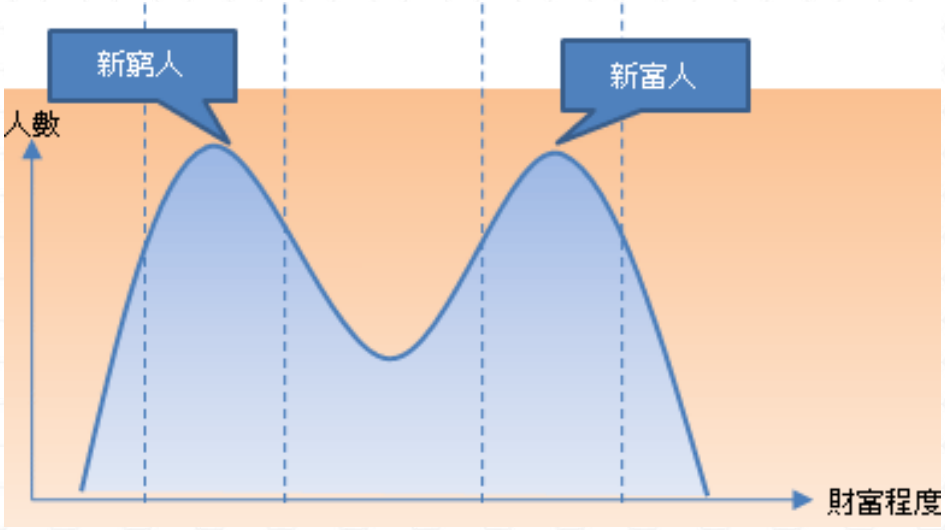
This book has pretty decent contents and well curated examples. But the serious problem is that there's no navigating bar supported for Mac OsX. I have contacted the customer service in many emails but circulating discussion, which made me send the screenshot for the diagnostic several times.

Missing the navigating bar is not tiny one as you need to find information yourself and back to the page in that you don't know where you are reading now. The publisher should fix this issue.

▸ 2 comments | 2 people found this helpful. Was this review helpful to you? Yes No Report abuse

過往了解使用者的反應

- 讓使用者評分
- 用五星量級評分
- 問卷調查
- 舉辦焦點團體



當抽樣方法有誤，所得到推論也有誤

搜尋引擎找出跟流病的關係

- 當感冒的人多了，使用 Google 查詢「發燒」或「咳嗽」的民眾也跟著變多，讓**特定關鍵字的搜尋熱門度，成了疫情變化的指標**

Google 的預測

- 預測的數值和真實的病情呈現超高度的正相關（相關係數高達 0.85）
- https://www.google.org/flutrends/about/

正相關係數（介於1~0之間）	等級
≧0.8	超高度相關（excellent correlation）
0.8~0.6	高度相關（good correlation）
0.6~0.4	中度相關（moderate correlation）
<0.4	低度或無相關（poor correlation）

使用Google Trend

- https://www.google.com.tw/trends/

但你又沒有Google 後面的資料

社群媒體的興起


facebook
電子郵件或電話
密碼
登入
忘記帳號？
Facebook，讓你和親朋好友保持聯繫，隨時分享生活中的每一刻。
註冊
永遠免費！
姓氏
名字
手機號碼或電子郵件
重新輸入手機號碼或電子郵件地址
新密碼
生日
年
月
日
為什麼需要提供出生日期的資料？
女性
男性
一旦點擊註冊，即表示你同意使用條款，而且你也閱讀了資料政策，包括 Cookie 的使用。
註冊


註冊
Log In
看看此刻正在發生什麼事。
尋找與你所喜愛事物相關的社群、對話和靈感。
精選
新聞時事
名人
影視
音樂
體育
更多
大中華
演藝
JJ Lin
@JJ_Lin
24 小時
你曾經好奇林俊傑的創作過程？
也好奇他的靈感都從哪裡來？
走進他的音樂，聽見林俊傑
這是一段 JJ 生命中最重要的歷程
也是一部關於音樂與人生的紀錄片
獻給所有音樂人以及熱愛音樂的人
.... fb.me/5JHPZmbeT
台灣蘋果日報 Taiwan News
@TW_nextmedia
22 小時
「少時」 #李灝 抵台賣 #彩妝 上百粉絲接機
新加入 Twitter?
立即註冊，取得你的個人化時間軸！
註冊
大中華


批踢踢實業坊
看板 Gossiping
精華區
最舊
‹ 上頁
下頁 ›
最新
4 [閒聊] L4D2是最經典的絕殺射遊戲嗎?
7/06 heil3266
1 Re: [新聞] 加班29.5小時只給3小時加班費 高雄銀行
7/06 putpi2007
Re: [新聞] 談大巨蛋爭議 柯：法律保護聘得起律師的人
7/06 unclefucka
Re: [閒聊] 人類以外的動物交配會爽嗎？
7/06 Meursault
! [公告] 八卦板板規(2016.02.16)
2/16 seabox
爆 M [爆卦] 最近發基法爭議整理
6/28 superlighter
56 Fw: [協尋]請大家幫我找大伯公~桃園新屋鄉埔頂村(代PO)
7/02 kun0616
爆 M [公告] 七月份置底閒聊文
7/01 Bignana
本網站已依台灣網站內容分級規定處理。此區域為限制級，未滿十八歲者不得瀏覽。

使用輿情分析了解民意

觀測清單

- PTT
- Facebook粉絲團
- 網路新聞媒體

每五分鐘可即時蒐集所有頻道評論

- 蘋果日報、時報資訊、NOW News、聯合新聞網、TVBS、中央通訊社、中廣新聞網、鉅亨網、新頭殼、民報、風傳媒、優活新聞網、健康醫療網、實況新聞網、Match 生活網、自由時報

每日統計報表

首頁　關鍵字　監控網頁　反向追蹤

設定查詢條件

起始日期

2014-10-01

終止日期

2014-11-15

監控來源

ptt

排序

date

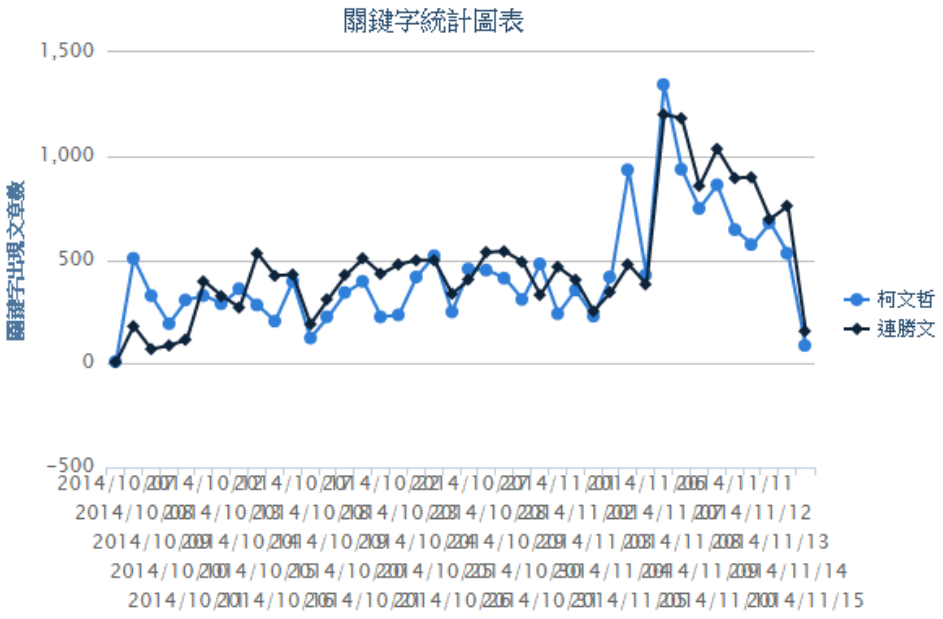
設定關鍵字

× 連勝文　× 柯文哲　× MG149

× 台北市長選舉

查詢

統計圖表



檢視關聯文章

關聯文章

[討論] 如果神豬被慘電，吱吱還能
HatePolitics comments:5 likes:4 2014/11/15

Re: [新聞] 支持者若不投票 連：以後遇捷運殺人只能
Gossiping comments:2 likes:3 2014/11/15

Re: [新聞] 零安樂死流浪狗安置地 連勝文：雲林、嘉
PublicIssue comments: likes:1 2014/11/15

[新聞] 連勝文：藉外界抹黑訓練抗壓能耐
HatePolitics comments:3 likes:1 2014/11/15

Re: [心情] 政治獻金--柯文哲做到了
HatePolitics comments:3 likes:2 2014/11/15

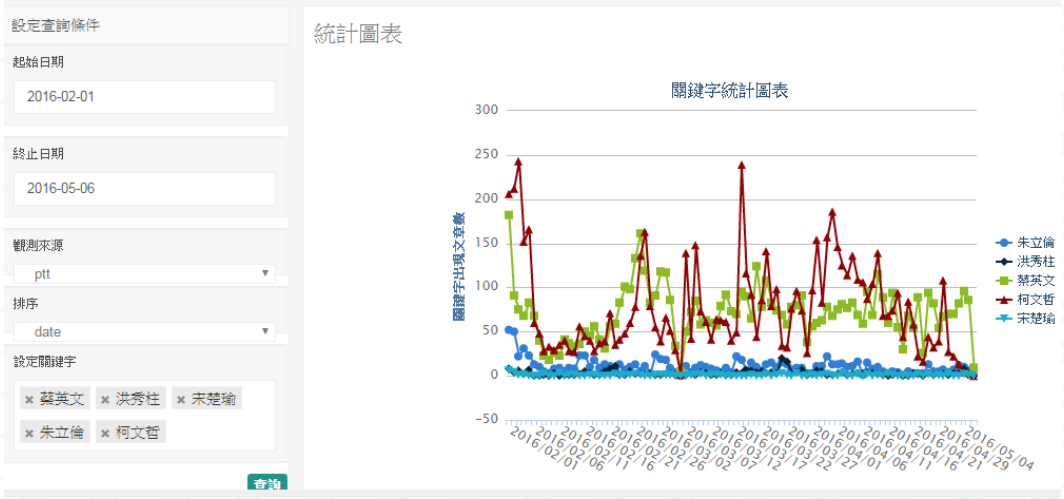
Re: [問卦] 有沒有連勝文選輸後下一步的八卦
Gossiping comments:3 likes:12 2014/11/15

看聲量說故事?

引用自壹週刊 2015/09/02 封面故事:
http://www.nextmag.com.tw/magazine/news/20150902/25253358

文字探勘

- 傳統的資料分析著重於結構化的資料
- 文字探勘的重點在於如何從非結構化的文字中，萃取出有用的重要資訊或知識
- 普遍應用
  - 民意調查
  - 事件追蹤
  - 找出關聯議題
  - 找出文章正負評
  - 文章摘要 (Summly)



文字探勘步驟

文字處理

- 斷詞
- 斷句

資料量化

- 詞頻計算
- 文字矩陣
- 計算TF-IDF

探勘分析

- 文字雲
- 文章分群
- 文章分類
- 關聯分析

中文斷詞技術

字串比對的斷詞方法

- 字串比對的斷詞方法
  - 將待分析的漢字串與詞典中的詞進行比對，若在詞典中找到某個字串，則比對成功
  - 按照比對方向的不同，比對斷詞方法可以分為正向和逆向
  - 按照不同長度優先比對的情況，可以分為最長優先和最短優先
  - 按照是否與詞性標注過程相結合，又可以分為單純斷詞方法和斷詞與標注相結合的方法

基於語意的斷詞方法

- 基於語意的斷詞方法
  - 斷詞的同時進行句法、語義分析，利用句法資訊和語義資訊來處理歧義現象
  - 模擬人對句子的理解過程，以根據詞、句子等的句法和語義資訊來對斷詞歧義進行判斷

基於統計的斷詞方法

- 基於統計的斷詞方法
  - 在內文(Context)中，相鄰的字同時出現的次數越多，就可能構成單詞
  - 根據隱藏馬可夫模型(Hidden Markov Model)所建立的斷詞系統
  - 根據條件隨機域 (CRF)所建立的斷詞系統

n-gram 方法屬於基於統計的斷詞方法

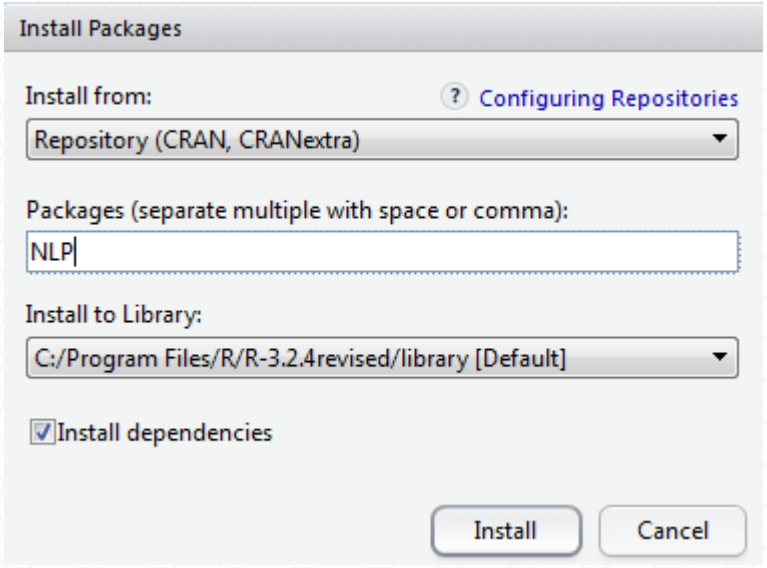
如何找出有意義的詞彙?

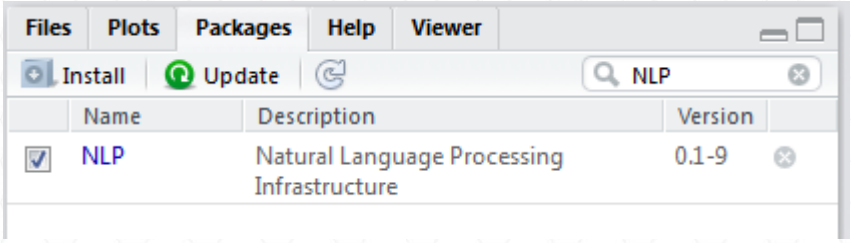
- 使用n-gram 做中文斷詞
- 假設n = 2
  - 統計所有 2-gram 的出現次數
  - 可表示成機率：出現次數除以總次數。
- n-gram 的缺點
  - 沒有參考中文文法

酸民婉君也可以報名嗎 → 酸民 民婉 婉君 君也 也可 可以 以報 報名 名嗎

安裝與讀取NLP 套件

```
install.packages("NLP")
library(NLP)
```





產生 bi-gram (2-gram)

```
s <- strsplit(x="那我們酸民婉君也可以報名嗎", split ='')
bigram <-ngrams(unlist(s), 2)
vapply(bigram, paste, "", collapse = "")
```

```
[1] "那我" "我們" "們酸" "酸民" "民婉" "婉君" "君也" "也可"
[9] "可以" "以報" "報名" "名嗎"
```

產生 tri-gram (3-gram)

```
s <- strsplit(x="那我們酸民婉君也可以報名嗎", split ='')
trigram <-ngrams(unlist(s), 3)
vapply(trigram, paste, "", collapse = "")
```

```
[1] "那我們" "我們酸" "們酸民" "酸民婉" "民婉君" "婉君也"
[7] "君也可" "也可以" "可以報" "以報名" "報名嗎"
```

找出出現兩次以上的詞

```
article <- "身兼中華職棒聯盟會長的國民黨立委吳志揚今天透露，台灣積極爭取的2017世界棒球經典賽分區預賽主辦權，因為大巨蛋遲遲無法孵出來，確定被競爭對手韓國拿走。吳志揚批評，當初中央政府拿台北市的精華地跟北市府交換，就是希望在大巨蛋現址成立體育園區，就如果北市府要改變使用目的，教育部都漠不關心，部長跟體育署長乾脆下台。"
w <- strsplit(x=article, split ='')
bigram <-ngrams(unlist(w), 2)
bigram.str <- vapply(bigram, paste, "", collapse = "")
tb <- table(bigram.str)
tb[tb>=2]
```

```
，就 大巨 北市 巨蛋 市府 吳志 志揚 體育
   2    2    3    2    2    2    2    2
```

如何斷句

```
strsplit(article, "、|，|。")
```

```
[[1]]
 [1] "身兼中華職棒聯盟會長的國民黨立委吳志揚今天透露"
 [2] "台灣積極爭取的2017世界棒球經典賽分區預賽主辦權"
 [3] "因為大巨蛋遲遲無法孵出來"
 [4] "確定被競爭對手韓國拿走"
 [5] "吳志揚批評"
 [6] "當初中央政府拿台北市的精華地跟北市府交換"
 [7] "就是希望在大巨蛋現址成立體育園區"
 [8] "就如果北市府要改變使用目的"
 [9] "教育部都漠不關心"
[10] "部長跟體育署長乾脆下台"
```

斷句後再做n-gram

```
a.split <- strsplit(article, "、|，|。")
w.split <- strsplit(x=unlist(a.split), split ='')

bigram <- function(w){
  bigram <-ngrams(unlist(w), 2)
  bigram.str <- vapply(bigram, paste, "", collapse = " ")
  bigram.str
}
bigram.all <- sapply(w.split, bigram)
tb <- table(unlist(bigram.all))
tb[tb>=2]
```

大巨 2
巨蛋 2

=

大巨蛋 2

大	巨	北	市	巨	蛋	市	府	吳	志	志	揚	體	育
	2		3		2		2		2		2		2

長詞優先法

- 最普遍被廣泛使用的斷詞方法
- 從句子的一端開始，取最長的詞串逐一比對辭典內的詞，若找到就把它當作斷詞的結果，再把句子中比對到的詞去除，剩下的部份再重複剛剛的動作，直到整句都斷詞完畢
- 通常若是有夠大的辭典，長詞優先法的正確率可高達90%

長詞優先演算法

- 給定一個連續句子S
- 給定常用詞典D
- 從最長詞n = 4到 n=2:
  - 從左到右掃描S
    - 檢查S中是否有關鍵詞在D中
      - 如是則移除該關鍵詞
    - 回傳移除關鍵詞的句子s’
  - 用n gram 將s’斷句
    - 將出現超過最小閥值的字加到字典D

移除關鍵字

```
s = "當初中央政府拿台北市的精華地跟北市府交換"
s.split = strsplit(s, '台北市')
paste(unlist(s.split), collapse = "", sep="")
```

"當初中央政府拿的精華地跟北市府交換"

建立移除關鍵字函式

```
removekey <- function(s, keys){
  for (key in keys){
    s.split = strsplit(s, key)
    s = paste(unlist(s.split), collapse = "", sep="")
  }
  s
}
removekey("當初中央政府拿台北市的精華地跟北市府交換", c("台北市", "中央"))
```

建立 ngram 斷詞函式

```
ngram.func <- function(w, n){
  n.gram <-ngrams(unlist(w), n)
  n.gram.str <- vapply(n.gram, paste, "", collapse = "")
  n.gram.str
}
```

實作長詞優先斷詞

長詞優先斷詞

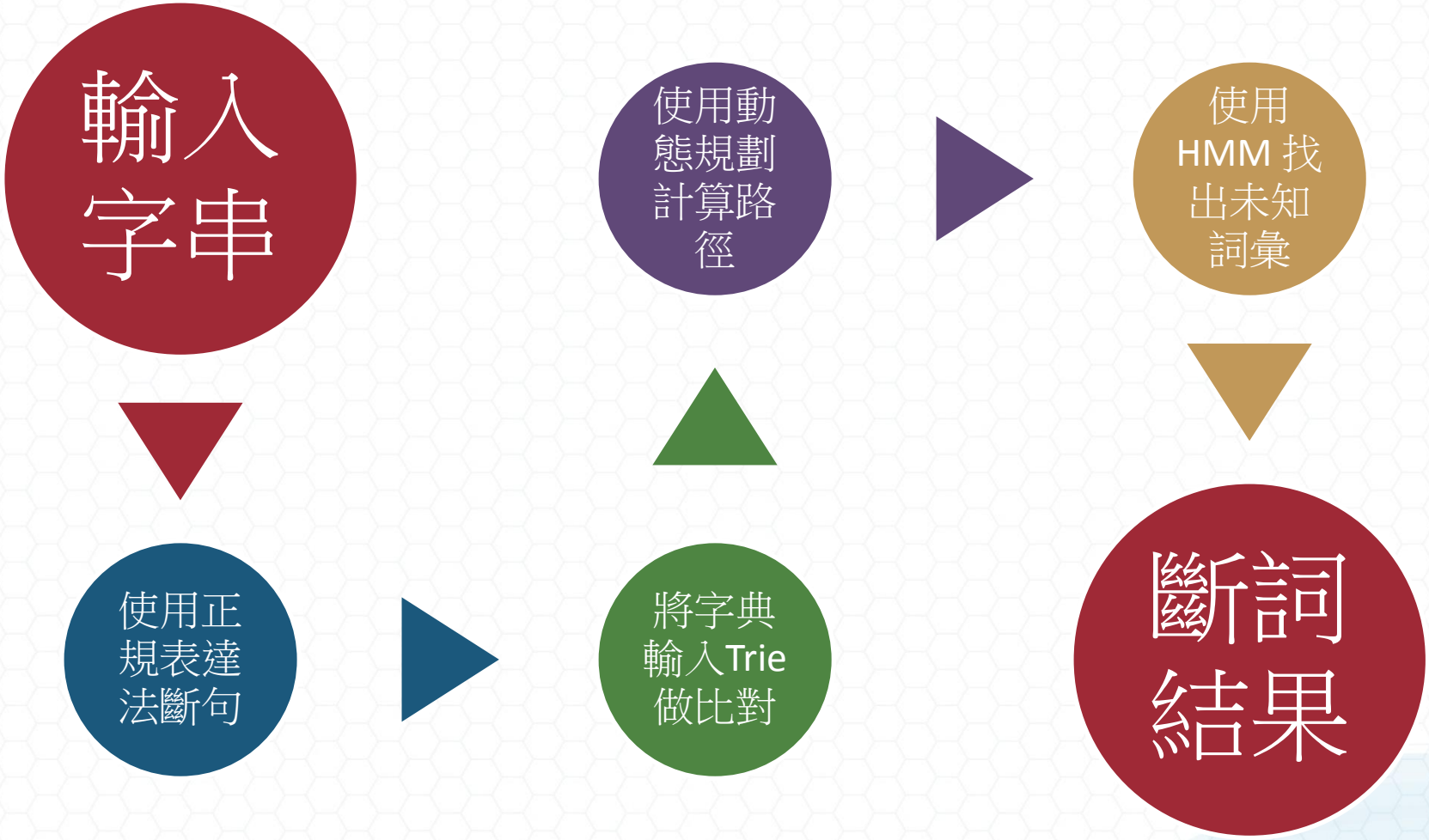
```
longTermFirst <- function(article, keywords){
  for(i in seq(4,2,-1)){
    article = removekey(article, keywords)
    a.split <- strsplit(article, "、|，|。")
    w.split <- strsplit(x=unlist(a.split), split ='')
    n.gram.all <- sapply(w.split, function(e) ngram.func(e,i))
    tb <- table(unlist(n.gram.all))
    candidate <- names(tb[tb>=2])
    keywords = c(keywords, candidate)
  }
  keywords
}

keywords = c()
longTermFirst(article, keywords)
```

依每句斷詞

出現超過兩次的列為關鍵字

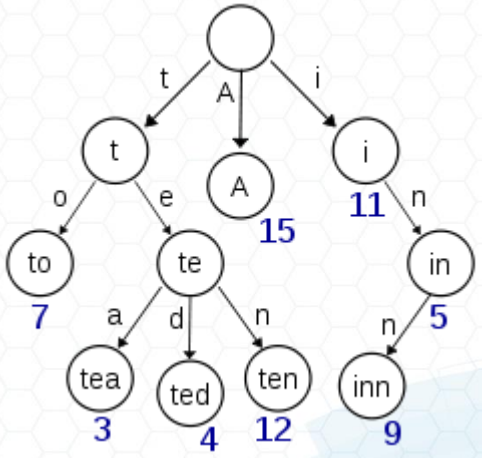
使用jiebaR

JiebaR 演算法


輸入字串
使用正規表達法斷句
將字典輸入Trie做比對
使用動態規劃計算路徑
使用HMM 找出未知詞彙
斷詞結果

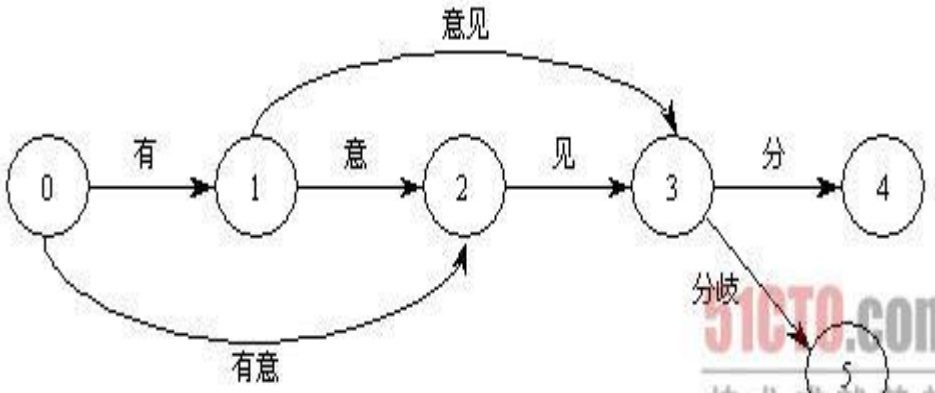
將字典輸入Trie 做比對

- Jieba 內建dict.txt的詞典, 裡面有2萬多條詞, 包含了詞條出現的次數和詞性
- 把這2萬多條詞語, 放到一個trie樹中, 而trie樹是有名的首碼樹, 也就是說一個詞語的前面幾個字一樣, 就表示他們具有相同的首碼, 就可以使用trie樹來存儲, 具有查找速度快的優勢



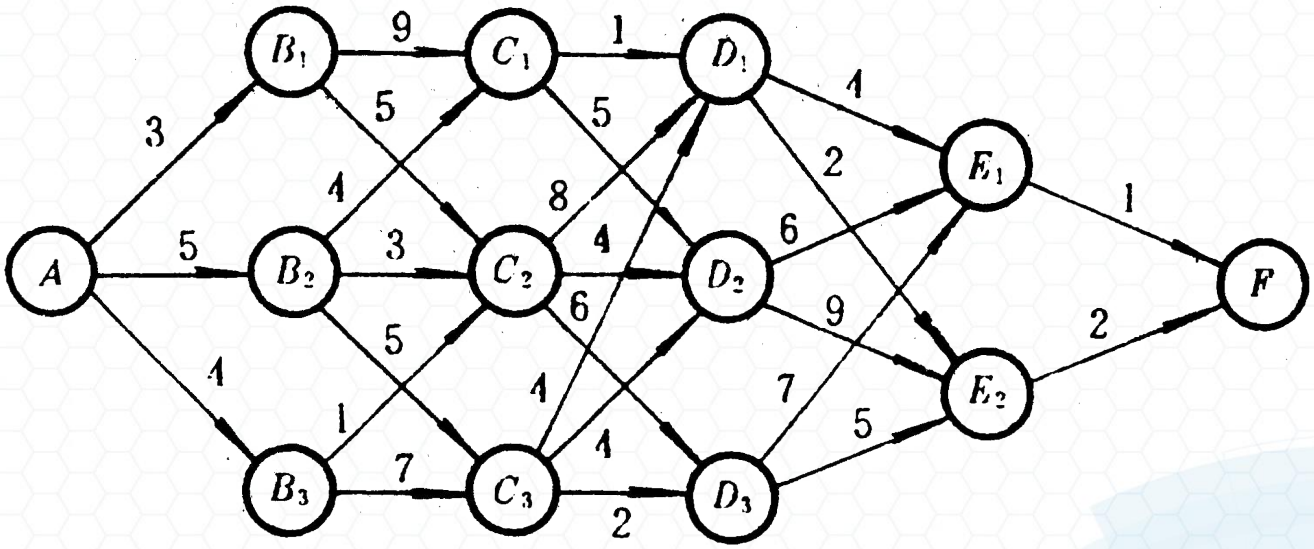
生成DAG

- 根據dict.txt生成trie樹
- 根據trie樹, 生成DAG根據比對到的字樣，產生幾種可能的端詞方法



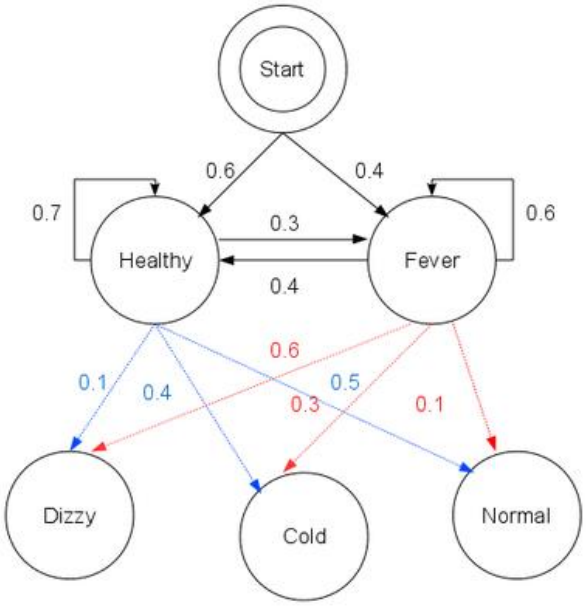
使用動態規劃計算路徑

- 動態規劃中, 先查找待句子中已經切分好的詞語, 對該詞語查找該詞語出現的頻率(次數/總數)
- 然後根據動態規劃查找最大機率路徑的方法, 對句子從右往左反向計算最大機率得到最大機率路徑的組合.



隱馬爾可夫模型

- 用來描述一個含有隱含未知參數的馬爾可夫過程
- 目的是從可觀察的參數中確定該過程的隱含參數。然後利用這些參數來作斷詞



誰是馬可夫?

- Andrey Markov

(14 June 1856 N.S. – 20 July 1922)

- Calculated letter sequences of the Russian language

問題描述

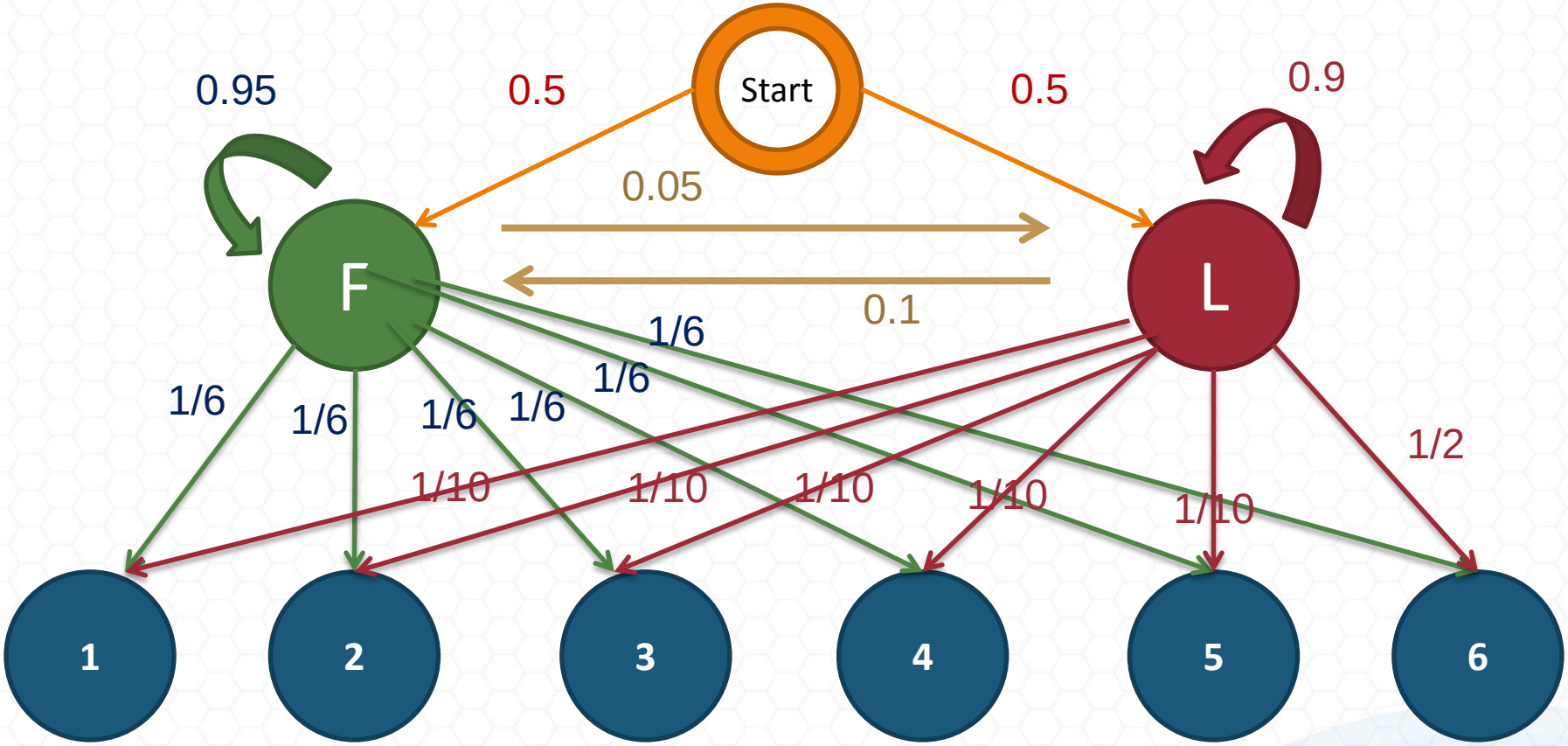
- States -> "F", "L“
- Transition Matrix

	Fair	Loaded
Fair	0.95	0.05
Loaded	0.1	0.9

- Emission Matrix

	Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	½

Problem Description



切詞範例

- 柯文哲相關資訊與新聞
  - 輸出的狀態序列為
  - BMEBEBESBE
- 可以切詞為
  - BME/BE/BE/S/BE
  - 柯文哲/相關/資訊/與/新聞
- **B後面只可能接(M or E)，不可能接(B or S)，而M後面也只可能接(M or E)，不可能接(B, S)**

機率矩陣

- 初始機率 InitStatus

```
#B -0.26268660809250016
#E -3.14e+100
#M -3.14e+100
#S -1.465263339853767 8
```

趨近於0

- 轉移矩陣機率 TransProbMatrix

	B	E	M	S
B	-3.14e+100	-0.510825623765990	-0.916290731874155	-3.14e+100
E	-0.5897149736854513	-3.14e+100	-3.14e+100	-0.8085250474669937
M	-3.14e+100	-0.33344856811948514	-1.2603623820268226	-3.14e+100
S	-0.7211965654669841	-3.14e+100	-3.14e+100	-0.6658631448798212

EmitProbMatrix 矩陣

- P(Observed[i], Status[j]) = P(Status[j]) * P(Observed[i]|Status[j])

```
#B
    柯 PB1, 文PB2 哲PB3
#E
    柯 PE1, 文PE2 哲PE3
#M
    柯 PM1, 文PM2 哲PM3
#S
    柯 PS1, 文PS2 哲PS3
```

從既有詞組發現每個單字出現的機率

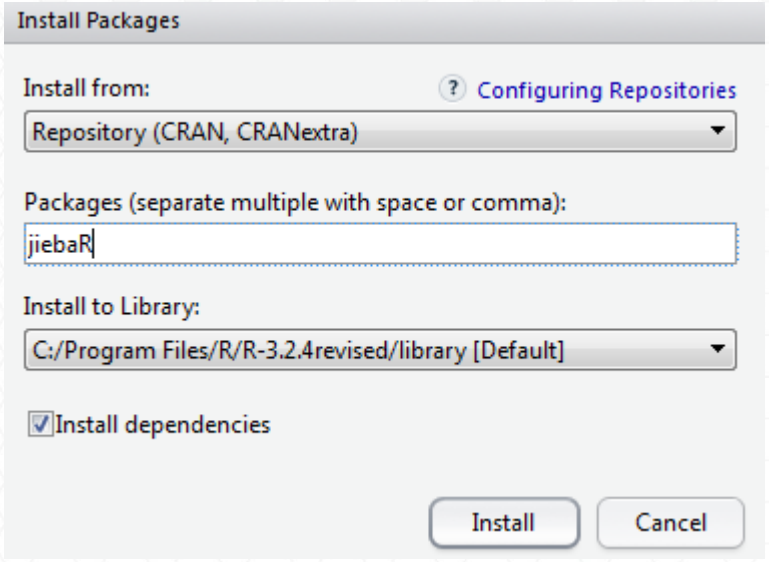
求出可能路徑

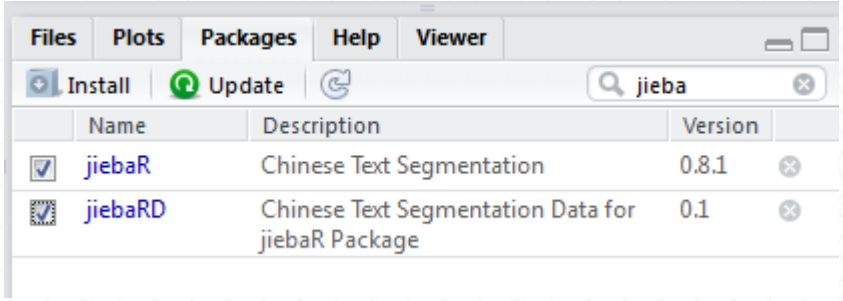
- EBSEBEBEMB
- 倒回來變成
  - BMEBEBESBE
- 可以切詞為 BME/BE/BE/S/BE
  - 柯文哲/相關/資訊/與/新聞

安装jiebaR

install.packages(“jiebaR")

library(jiebaR)





使用jiebaR 斷詞

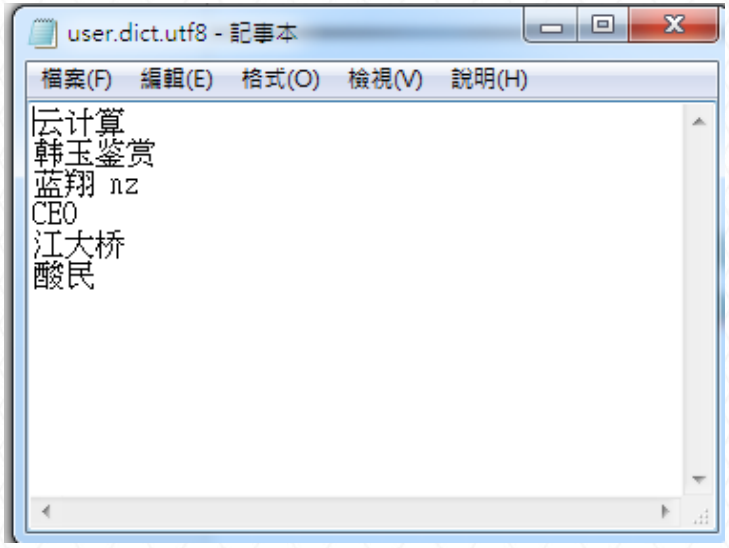
```
s="那我們酸民婉君也可以報名嗎"
mixseg = worker()
segment(code= s , jiebar = mixseg)
```

那我們酸民婉君也可以報名嗎

```
[1] "那"      "我們"    "酸民婉君" "也"      "可以"    "報名"    "嗎"
```

編輯使用者自定義字典

```
edit_dict()
USERPATH
[1] "C:/Program Files/R/R-3.2.4revised/library/jiebaRD/dict/user.dict.utf8"
```



AA制 3 n
AB型 3 n
AT&T 3 nz

字典範例：字詞、詞頻、詞性

抓出詞性

```
tagseg = worker('tag')
segment(s, tagseg)
```

```
  r     r     x     x     d     c     v     y
"那" "我們" "酸民" "婉君"  "也" "可以" "報名"  "嗎"
```

可以在字典中加註詞性

酸民 n

詞性說明
https://gist.github.com/luw2007/6016931

中文詞庫擴充

- 維基百科
  - https://zh.wikipedia.org/zh-tw/Wikipedia:%E9%A6%96%E9%A1%B5
- 萌典
  - https://www.moedict.tw/~%E4%B8%96%E7%95%8C

抓取關鍵詞

```
key = worker('keywords', topn = 3)
key <= s
11.7392 11.7392 11.7392
 "報名"  "婉君"  "我們"
```

使用TFIDF判斷關鍵詞

找出文章關鍵詞

- 如何判斷一個詞是不是**關鍵詞**
  - 如果某個詞比較少見，但是在這篇文章中多次出現，那麼該詞很反映了這篇文章的特性
  - 可用來評估該詞對於該文件的重要程度
  - 使用TF * IDF，假設單詞對文章的重要性越高，TF-IDF值就越大

TF-IDF

- TF (Term Frequency)
  - 單詞在該文件的出現次數
  - 單詞w在文檔d中出現的次數: count(w, d)
  - 文檔d中總詞數: size(d)
  - tf(w,d) = count(w, d) / size(d)
- IDF (Inverse Document Frequency)
  - 一個詞語普遍重要性的度量
  - 設文檔總數為n
  - 設詞w所出現檔數docs(w, D)
  - idf = log(n / docs(w, D))

計算TF-IDF

```
a <- c("a")
abb <- c("a", "b", "b")
abc <- c("a", "b", "c")
D <- list(a, abb, abc)

tfidf <- function(t,d, D){
 tf <- table(d)[names(table(d)) == t]/ sum(table(d))
 idf <- log(length(D) /sum(sapply(D, function(e) t %in% e)))
 tf*idf
}
```

計算字詞在文章的重要性

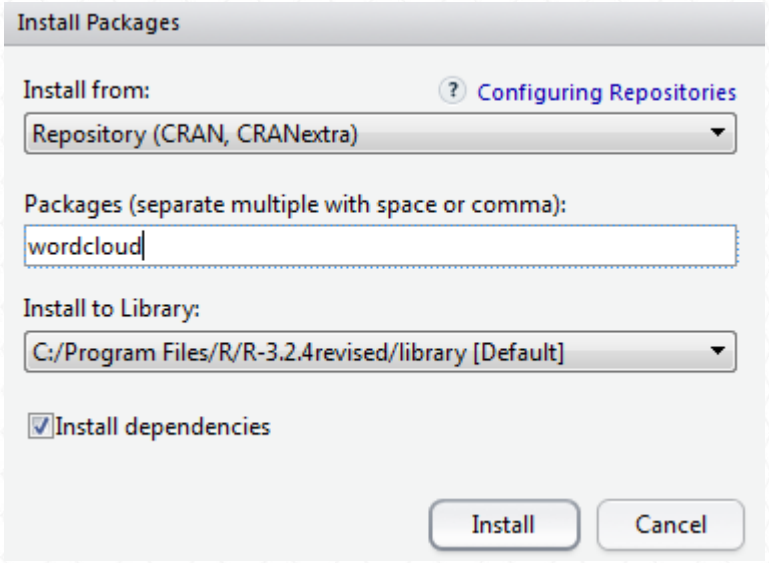
```
> tfidf('a',a,D)
a
0
> tfidf('b',abb,D)
        b
0.2703101
> tfidf('b',abc,D)
       b
0.135155
> tfidf('c',abc,D)
        c
0.3662041
> tfidf('b',abc,D)
       b
0.135155
```

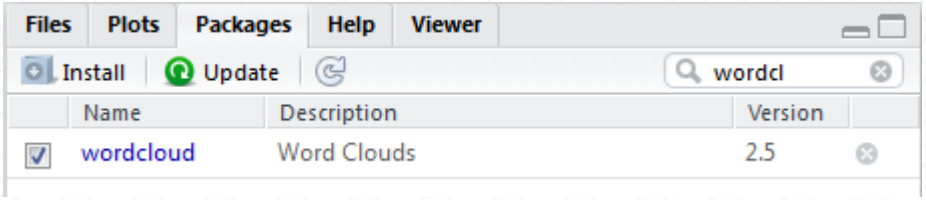
計算詞頻

```
word <- unlist(sapply(appledaily$content,
function(e) segment(code= as.character(e) , jiebar
= mixseg)))
tb <- table(word)
```

安装wordcloud

```
install.packages("wordcloud")
library(wordcloud)
```

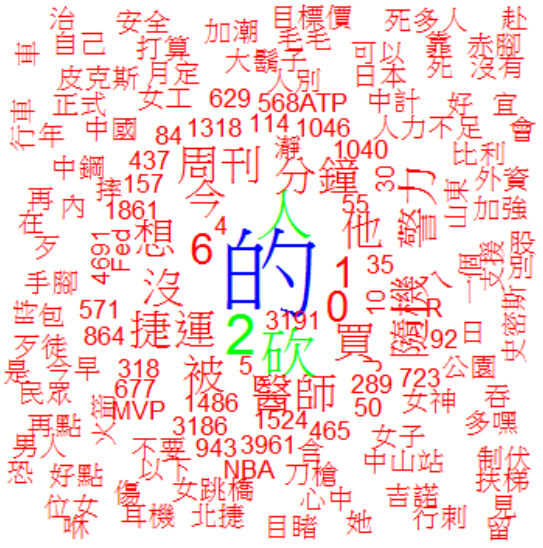




呈現文字雲

```
library(wordcloud)
wordcloud(names(tb), tb, , min.freq = 1,
random.order = F, ordered.colors = F, colors =
rainbow(length(1:3)))
```

可能要篩選一下詞彙



THANK YOU