中文文字資料探勘

李艦

檬果資料分析技術（上海）有限公司

2013.12.06

目 录

相关概念

- 文字資料探勘
  - 文字資料探勘（Text mining），也被稱為文本挖掘。一般而言，指的是從非結構化的文字中，萃取出有用的重要資訊或知識。通常会采用自然语言处理的方法进行分析。
- 自然語言處理
  - Natural Language Processing，簡稱NLP，是人工智慧和語言學領域的分支學科。在這此領域中探討如何處理及哂米匀徽Z言。通常包括詞性標注、文本分類、自動摘要、信息檢索、中文斷詞等研究範疇。

big data 时代的Text mining

- 網路架構的升級
- 社群網路服務的興起
- 電子商務的成熟
- 語音識別技術的廣泛應用

Why use R?

- prototype
  - 資源豐富、易於學習；
  - 非常靈活，適合開發新的方法；
  - 可以不用考虑計算效率。
- 工程化開發
  - 開發時間很短；
  - 易於集成到穩定的系統中；
  - 可以利用其它工具提高穩定性和計算的性能。

分析框架

- tm
  - 最通用的框架，被幾乎所有的NLP類包引用。
  - tm.plugin.dc、tm.plugin.mail、tm.plugin.factiva 是針對tm包的擴展，可以用來分佈式存儲語料、處理郵件文本、獲取Factiva語料。
  - RcmdrPlugin.temis 提供了命令行工具。
- openNLP
  - Apache OpenNLP 的R軟體接口；
  - 可以進行單句識別、句分解、句結構分析、構建語法樹等；
  - 相對比較底層，一般的文字資料探勘任務需要在該包基礎上進行二次開發。中文支持不是很好。
- qdap
  - 一個綜合了定量分析和定性分析的工具包;
  - 包含一些自然語言處理的相關函數。
- koRpus
  - 綜合的文本分析的包，詞頻分析居多;
  - 可讀性分析和語種識別比較有特色。

詞分析

- 關鍵詞提取
  - 通過訓練自動提取文檔中的關鍵詞;
  - RKEA包提供了KEA的接口可以用來進行關鍵詞提取。
- 詞雲
  - wordcloud包使用原生的R語言繪製詞雲;
  - 該包只能在本地字符環境下使用，字符編碼上存在缺陷。
- 詞頻分佈
  - zipfR提供了一些關於詞頻分佈的統計模型，尤其是詞頻分佈中最常用的齊普夫定律。
- 其他語言
  - wordnet的包提供了一個英文文本數據庫的接口，KoNLP是一個韓文自然語言處理的包。
  - Snowball、SnowballC、Rstem 是進行詞幹提取的包。

語義分析Semantics

- Topic Model
  - 自動識別不同Topic，並提取各Topic的關鍵詞；
  - topicmodels包提供了C接口使用LDA和相关Topic模型来建模。lda包是lda模型的另一种实现。
- 文本聚類和分類
  - RTextTools包專門用來進行自動文本分類。skmeans包提供了幾種模糊k均值的算法。textcat包可以進行基於n元語法短語的文本聚類。movMF 提供了一種基於概率模型（基於vMF分佈）的文本聚類方法。
- 潛語義分析
  - 通過對文檔詞條矩陣進行奇異值分解來降維，然後計算相似度。lsa包可以用來進行分析。
- 綜合分析
  - kernlab包，提供了一些核機器學習的方法進行文本分類，聚類，新穎性檢測降維等。
  - textir包提供了一些函數進行文本和語義挖掘。

字符處理

- 内置字符函數
  - help.search(keyword = “character”, package = “base”)
- 字符編碼
  - Encoding 和iconv
  - tau 包
- 正則表達式（Regular Expression）
  - grep和sub系列函數
  - gsubfn包
- 擴展字符處理
  - stringr 包

其他工具

- Rweibo
  - 利用新浪API通過OAuth的方式獲取微博信息，另外提供了使用RCurl和XML解析網頁獲取數據的函數。

```
install.packages("Rweibo",
    repos = "http://R-Forge.R-project.org")
```

- Rwordseg
  - 中文斷詞包，引用了基於Java的Ansj斷詞工具，使用隱馬爾可夫模型進行斷詞。

```
install.packages("Rwordseg",
    repos = "http://R-Forge.R-project.org")
```

目 录

Rwordseg包实现中文斷词

```
segmentCN("中華R軟體學會")

## [1] "中華" "R"    "軟體" "學會"

segmentCN("中国R语言协会", nature = TRUE)

##      ns       n       n
## "中国" "R语言" "协会"

segmentCN("中華R軟體學會", returnType = "tm")

## [1] "中華 R 軟體 學會"

segmentCN("D:\\说岳全传_GBK.txt")

## Output file:  D:\说岳全传_GBK.segment.txt
## [1] TRUE
```

新詞加入與人名識別

```
segmentCN("可变焦，长焦和微距效果都很好")
```

```
##  [1] "可变" "焦"   "长"   "焦"   "和"   "微"   "距"
##  [8] "效果" "都"   "很"   "好"
```

```
insertWords(c("长焦", "微距", "可变焦"))
segmentCN("可变焦，长焦和微距效果都很好")
```

```
##  [1] "可变焦" "长焦"   "和"     "微距"   "效果"
##  [6] "都"     "很"     "好"
```

```
segment.options(isNameRecognition = TRUE)
segmentCN("可变焦，长焦和微距效果都很好")
```

```
##  [1] "可变焦" "长"     "焦和微" "距"     "效果"
##  [6] "都"     "很"     "好"
```

詞典管理

```
installDict("D:\\tw.dic")

## OK!
## New dictionary was installed, please restart R
## to use it.

dic1 <- importSogouScel("D:\\金庸武功招式.scel")
listDict()

##          Name Type     Des
## 1 userDefine       tw.dic
## 2 userDefine       金庸武功招式.scel

uninstallDict()

## OK!
## The user defined dictionary was uninstalled,
## please restart R.
```

Word cloud


阳性
发现
大家
市场
新闻
复核
广东省
东莞市
我们
自己
控制
流感
批发
广东
国家
没有
什么
动物
实验室
发布
可以
东莞
东城
接触
感染
猪肉
今天
预防
病毒
中国
样品
中心
参考
患者
农业部
上海

建立語料對象

- 所有的原始文本都必須存成語料對象

```
d.corpus <- Corpus(VectorSource(d.vec))
d.corpus

## A corpus with 1583 text documents

inspect(d.corpus[1])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs
## and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## [[1]]
## 跟 一起 奋斗 别 再 吃 白切鸡 了 今日 惠州 新闻
链接 东莞市 东城 三 鸟 批发 市场 鸡 样品 确认 为
阳性
```

通過tm內置的機制來處理語料對象

- 使用tm_map 函數將某個處理語料的函數傳入；
- tm包沒有中文的停止詞，可以使用tmcn包中的stopwordsCN函數。

```
d.corpus <- tm_map(d.corpus, removeWords, stopwordsCN())
```

建立文檔詞條矩陣（document-term matrix）

- 文檔詞條矩陣是整個tm包乃至現階段所有R軟體文本挖掘相關的包的最基礎對象。

```
d.dtm <- DocumentTermMatrix(d.corpus)
d.dtm
```

```
## A document-term matrix (1583 documents, 1748 terms)
##
## Non-/sparse entries: 4319/2762765
## Sparsity           : 100%
## Maximal term length: 15
## Weighting          : term frequency (tf)
```

利用文檔詞條矩陣進行文本分析

- 例如查找頻数超過100的詞以及和某個詞的關聯度超過0.5的詞

```
findFreqTerms(d.dtm, 100)

## [1] "实验室" "禽流感"

findAssocs(d.dtm, "实验室", 0.5)

## 办公室 东莞市 禽流感
##   0.54   0.53   0.51
```

利用文檔詞條矩陣進行文本分析

- Topoc Model

```
library(topicmodels)
ctm <- CTM(d.dtm.sub2, k = 2)
terms(ctm, 2, 0.1)
```

```
## $`Topic 1`
## [1] "东莞市" "农业部" "实验室"
##
## $`Topic 2`
## [1] "禽流感"
```

tm包的缺點

- 中文支持不是很好
  - 没有採用UTF-8的編碼方式，而是針對不同字符集進行處理，並沒有包含中文字符集的處理方式。
- 對象過於繁瑣但是封裝性不好
  - 所有數據結構都使用自定義的方式，需要其他函數來適應；
  - 基於S3的開發而不是S4，封裝性不好。
- 為大數據設計但是處理大數據的能力不強
  - 設計思想是針對大數據的文本挖掘，目前也存在一些第三方的分佈式計算支持；
  - 但是使用R進行文本分析的場景多半是實驗性質，很多靈活的方法不容易在tm包中實現。

提高tm 的性能

- 並行
  - 使用tm.plugin.dc
  - 分佈式存儲与MapReduce
  - 常用於語料庫的預處理
- 代數計算
  - 更換BLAS 和LAPACK;
  - 矩陣計算進行算法優化，例如奇異值分解。

目 录

tmcn包的安装

- 核心包

```
install.packages("tmcn",
    repos = "http://R-Forge.R-project.org")
```

- CRF++ 擴展包

```
install.packages("tmcn.crfpp",
    repos = "http://R-Forge.R-project.org")
```

- word2vec 擴展包[a]

```
install.packages("tmcn.word2vec",
    repos = "http://R-Forge.R-project.org")
```

[a]目前tmcn.word2vec 包的Windows 版本在R-Forge 下編譯有問題，請下載源碼自行編譯或者到作者主頁下載二進制版本。

tmcn包功能簡介及開發計劃

- 中文編碼
  - 各種編碼的識別和UTF-8之間的轉換；
  - 中文簡體字和繁體字之間的轉換；
  - 增強了tau 包中的一些功能。
- 中文語料資源
  - 例如GBK字符集及中文停止詞等。
- 字符處理
  - 常用的字符處理函數；
  - 一些函數是對stringr包的優化或者不同實現。
- Text Mining
  - 對tm 包進行補充，比如CRF++、word2vec等；
  - 基於基礎R對象的文本挖掘框架；
  - 高性能計算的實現。

GBK字符集

```
data(GBK)
head(GBK)
```

```
##   GBK py0          py Radical Stroke_Num_Radical
## 1  吖   a       ā yā       口                   3
## 2  阿   a     ā ɑ ē        阝                   2
## 3  啊   a ɑ á à ǎ ā        口                   3
## 4  锕   a         ā        钅                   5
## 5  錒   a         ā        金                   8
## 6  嗄   a     á shà        口                   3
##                 Stroke_Order Structure    Freq
## 1                 丨フ一丶ノ丨       左右      26
## 2               フ丨一丨フ一丨       左右  526031
## 3           丨フ一フ丨一丨フ一丨     左中右   53936
## 4       ノ一一一フフ丨一丨フ一丨     左中右       3
## 5 ノ丶一一丨丶ノ一フ丨一丨フ一丨       左右       0
## 6     丨フ一一ノ丨フ一一一ノフ丶       左右      11
```

字符編碼識別

```
txt1 <- "中華R軟體學會"
c(isUTF8(txt1), isGBK(txt1), isBIG5(txt1))

## [1] FALSE  TRUE FALSE

txt2 <- iconv(txt1, "GBK", "big5")
c(isUTF8(txt2), isGBK(txt2), isBIG5(txt2))

## [1] FALSE  TRUE  TRUE
```

UTF-8轉換

```
txt1 <- "R Taiwan 2013 研討會"
Encoding(txt1) <- "big5"
txt1

## [1]  "R Taiwan 2013 鐮旐◣鏈\x83"

toUTF8(txt1)

## [1] "R Taiwan 2013 研討會"

catUTF8(txt1)

## [1] R Taiwan 2013 \u7814\u8A0E\u6703

revUTF8("<U+7814><U+8A0E><U+6703>")

## [1] "研討會"
```

中文字符轉換

```
txt1 <- c("中国R语言会议")
toTrad(txt1)

## [1] "中國R語言會議"

toTrad("中國R語言會議", rev = TRUE)

## [1] "中国R语言会议"

toPinyin(txt1, capitalize = TRUE)

## [1] "ZhongGuoRYuYanHuiYi"
```

字符處理

```
txt1 <- c("\t(x1)a(aa2)a ", " bb(bb)")
strextract(txt1, "\\([^)]*\\)")

## [[1]]
## [1] "(x1)"  "(aa2)"
##
## [[2]]
## [1] "(bb)"

strstrip(c("\taaaa ", " bbbb    "))

## [1] "aaaa" "bbbb"
```

條件隨機場CRF

```
require(tmcn.crfpp)
TestFilePath <- system.file("tests",
package = "tmcn.crfpp")
WorkPath <- tempdir()
# Learn
TempletFile <- file.path(TestFilePath,
"testdata", "chunking_template")
TrainingFile <- file.path(TestFilePath,
"testdata", "chunking_train")
ModelFile1 <- file.path(WorkPath,
"output", "model1")
res1 <- crflearn(TempletFile, TrainingFile, ModelFile1)
# Test
KeyFile <- file.path(TestFilePath, "testdata",
"chunking_key")
ResultFile1 <- file.path(WorkPath, "output", "result1")
test1 <- crftest(res1$model_file, KeyFile, ResultFile1)
```

word2vec

```
require(tmcn.word2vec)
TestFilePath <- system.file("tests",
package = "tmcn.word2vec")
WorkPath <- tempdir()
TrainingFile1 <- file.path(TestFilePath,
"testdata", "questions-words.txt")
ModelFile1 <- file.path(WorkPath, "output", "model1.bin")
res1 <- word2vec(TrainingFile1, ModelFile1)
distance(res1$output_file, "think")[1:3, ]
```

```
##      Word   CosDist
## 1 vanish 0.9964207
## 2    walk 0.9954072
## 3    swim 0.9911700
```

開發相關事項

- 開發環境
  - 當前最新版本的tmcn包是0.1-2
  - 源代碼使用SVN方式管理，目前發布在 R-forge: https://r-forge.r-project.org/R/?group_id=1571，成熟後會發布到 CRAN。
  - 所有代碼在32位 Win7、64位 Win7 及64位 Ubuntu 12.04 進行測試
- To-Do List[a]
  - 完善文本挖掘中的各種模型和算法
  - 進一步優化該包中的函數
  - 建立一個能兼容tm 包的框架
  - 優化高性能的解決方案

[a]，本包及To-Do List会隨時更新，請關注R-forge上的開發主頁或者作者主頁http://jliblog.com/app/tmcn

Thank you!

李艦 Email: jli@mango-solutions.com