中文文本挖掘和 tmcn 包

李舰

檬果数据分析技术（上海）有限公司

第六届中国R语言会议（上海会场）

2013.11.03

目 录

分析框架

- tm
  - 最通用的框架，被几乎所有NLP类包引用。
  - tm.plugin.dc、tm.plugin.mail、tm.plugin.factiva 是针对tm 包的扩展，可以用来分布式存储语料、处理邮件文本、获取Factiva 语料。
  - RcmdrPlugin.temis 提供了命令行工具。
- openNLP
  - Apache OpenNLP 的R 语言接口；
  - 可以进行单句识别、句分解、句结构分析、构建语法树等；
  - 相对比较底层，一般的文本挖掘任务需要在该包基础上进行二次开发。中文支持不是很好。
- qdap
  - 一个综合了定量分析和定性分析的大杂烩；
  - 包含一些自然语言处理的相关函数。
- koRpus
  - 综合的文本分析的包，词频分析居多；
  - 可读性分析和语种识别比较有特色。

词分析

- 关键词提取
  - 通过训练自动提取文档中的关键词；
  - RKEA 包提供了KEA的接口可以用来进行关键词提取。
- 词云
  - wordcloud 包使用原生的R绘制词云；
  - 该包只能在本地字符环境下使用，字符编码上存在缺陷。
- 词频分布
  - zipfR提供了一些关于词频分布的统计模型，尤其是词频分布中最常用的Zipf定律。
- 其他语言
  - wordnet包提供了一个英文文本数据库的接口，KoNLP 是一个韩文自然语言处理的包。
  - Snowball、SnowballC、Rstem 是进行词干提取的包。

语义分析

- 主题模型
  - 自动识别不同主题，并提取各主题的关键词；
  - topicmodels包提供了C接口使用LDA和相关主题模型来建模。lda包是lda模型的另一种实现。
- 文本聚类和分类
  - RTextTools包专门用来进行自动文本分类。skmeans 包提供了几种模糊KMeans 的算法。textcat 包可以进行基于n-gram短语的文本聚类。movMF 提供了一种基于概率模型（基于vMF 分布）的文本聚类方法。
- 潜语义分析
  - 通过对文档词条矩阵进行奇异值分解来降维，然后计算相似度。lsa包可以用来进行分析。
- 综合分析
  - kernlab包，提供了一些核机器学习的方法进行文本分类、聚类、新颖性检测降维等。
  - textir包提供了一些函数进行文本和语义挖掘。

字符处理

- 内置字符函数
  - help.search(keyword = "character", package = "base")
- 字符编码
  - Encoding 和iconv
  - tau 包
- 正则表达式
  - grep和sub系列函数
  - gsubfn包
- 扩展字符处理
  - stringr 包

其他工具

- Rweibo
  - 利用新浪API通过OAuth的方式获取微博信息，另外提供了使用RCurl和XML解析网页获取数据的函数。

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

- Rwordseg
  - 中文分词包，调用了基于Java的Ansj分词工具，使用隐马尔可夫模型进行分词。

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

中文分词

- 设置returnType的参数可以直接输出tm接受的格式

```
d.vec <- segmentCN(text1, returnType = "tm")
length(d.vec)
```

```
## [1] 1583
```

```
d.vec[1]
```

```
## [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"
```

建立语料对象

- 所有的原始文本都必须存成语料对象

```
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
```

通过tm内置的机制来处理语料对象

- 使用tm_map 函数将某个处理语料的函数传入；
- tm包没有中文的停止词，可以使用tmcn 包中的stopwordsCN 函数。

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

建立文档词条矩阵

- 文档词条矩阵是整个tm 包乃至现阶段所有R语言文本挖掘相关的包的最基础对象。

```
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)
```

利用文档词条矩阵进行文本分析

- 例如查找频数超过100的词以及和某个词的关联度超过0.5的词

```
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
```

利用文档词条矩阵进行文本分析

- 例如主题模型

```
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包的缺点

- 中文支持不是很好
  - 没有采用UTF-8的方式，而是针对不同字符集进行处理，并没有包含中文字符集的处理方式。
- 对象过于复杂但是封装性不好
  - 所有数据结构都使用自定义的方式，需要其他函数来适应
  - 基于S3开发而不是S4，封装性不好
- 为大数据设计但不适合大数据
  - 设计思想是针对大数据的文本挖掘，目前也存在一些第三方的分布式运算的支持
  - 但是使用R进行文本分析的场景多半是实验性质，很多灵活的方法不是很容易在tm 包中实现

目 录

tmcn包的安装

- 核心包

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

- CRF++ 扩展包

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

- word2vec 扩展包[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 下编译有问题，请下载源码自行编译或者到作者主页下载二进制版本。

tmcn包功能简介及开发计划

- 中文编码
  - 各种编码的识别和UTF-8之间的转换
  - 中文简体字和繁体字之间的转换
  - 增强了tau 包中的一些功能
- 中文语料资源
  - 例如GBK字符集及中文停止词等
- 字符处理
  - 常用的字符处理函数
  - 一些函数是对stringr 包的优化或者不同实现
- 文本挖掘
  - 基于基础R对象的文本挖掘框架
  - 包含常用的文本挖掘模型
  - 包含一些独立的NLP库，比如CRF++、word2vec等。

GBK字符集

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

	GBK	py0	py	Radical	Stroke_Num_Radical	Stroke_Order	Structure	Freq
1	吖	a	ā yā	口	3	丨フ一丶ノ丨	左右	26
2	阿	a	ā ɑ ē	阝	2	フ丨一丨フ一丨	左右	526031
3	啊	a ɑ á à ǎ ā		口	3	丨フ一フ丨一丨フ一丨	左中右	53936
4	锕	a	ā	钅	5	ノ一一一フフ丨一丨フ一丨	左中右	3

字符编码识别

```
> txt1 <- c("\u4E2D\u56FD\u8BED\u8A00\u4F1A\u8BAE")
> txt2 <- iconv(txt1, "UTF-8", "GBK")
> txt3 <- txt1
> Encoding(txt3) <- "GBK"
> isUTF8(txt1)

[1] TRUE

> isGBK(txt2)

[1] TRUE

> isGBK(txt3)

[1] FALSE
```

UTF-8转换

```
> setCN()
> txt1 <- c("中国R语言会议")
> toUTF8(txt1)

[1] "中国R语言会议"

> catUTF8(txt1)

\u4E2D\u56FDR\u8BED\u8A00\u4F1A\u8BAE

> revUTF8("<U+4E2D><U+56FD>R<U+4F1A><U+8BAE>")

[1] "中国R会议"
```

中文字符转换

```
> 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"
```

字符处理

```
> 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"
```

条件随机场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 0.9964182
2    walk 0.9954073
3    swim 0.9911690
```

开发相关事项

- 开发环境
  - 当前最新版本的tmcn包是0.1-2
  - 源代码使用SVN方式管理，目前发布在 R-forge: https://r-forge.r-project.org/R/?group_id=1571，成熟后会发布到 CRAN，暂无计划迁往 GitHub
  - 所有代码在32位 Win7、64位 Win7 及64位 Ubuntu 12.04 进行测试
- To-Do List[a]
  - 完善文本挖掘中的各种模型和算法
  - 进一步优化该包中的函数
  - 建立一个能兼容 tm 包的框架
  - 提供高性能的解决方案

[a]，本包及To-Do List会随时更新，请关注R-forge上的开发主页或者作者主页http://jliblog.com/app/tmcn

Thank you!

李舰 Email: lijian.pku@gmail.com