Shiny 网页构建报告

龚政 PB20000053

开发背景

如今，在数据分析中，人们越来越注重分析结果的呈现，而词云图正是对文本内容的高度浓缩和精简处理，能更直观的反映特定文本的内容，直观明了，能节省读者时间，让读者在短时间内对文本数据的主要信息做到一目了然。

功能介绍

本shiny网页实现的功能汇总如下：

1. Word Cloud1部分：对于用户上传的“txt”格式全英文文件，打印出其对应的词云图，并能够控制该词云图中单词的最小频率和单词的最大重复次数，从而更加精确地找出我们想要的部分。
2. Word Cloud2部分：实现了对示例数据文件“demoFreq”的词云展示，用户可以控制该词云图的多种特性，即：字体，字体颜色，图片背景颜色，字体旋转角度，图像形状，字体大小，字体粗细，字体的旋转比例。

代码实现

准备工作

引入安装包

```
library(shiny)
library(tm)
library(wordcloud)
library(wordcloud2)
library(memoise)
```

`Word cloud 1`

全局函数`textProcess`

```
textProcess <- memoise(function(text) {

  myCorpus <- Corpus(VectorSource(text))
  myCorpus <- tm_map(myCorpus, content_transformer(tolower))
  myCorpus <- tm_map(myCorpus, removePunctuation)
  myCorpus <- tm_map(myCorpus, removeNumbers)
  myCorpus <- tm_map(myCorpus, removeWords, c("this", "that", "it", "the", "and",
"but"))

  myDTM = TermDocumentMatrix(myCorpus, control = list(minWordLength = 1))

  m = as.matrix(myDTM)

  sort(rowSums(m), decreasing = TRUE)
```

```
})
```

`TermDocumentMatrix` 函数：构建了一个描述了在文档中出现的单词的频率的数学矩阵。

将语料库中的文字依次进行小写化、去除标点、去除数字、去除常见无关单词的操作后，创建单词-频率矩阵`m`，并依频率降序排列。

`ui` 部分

```
fixedRow(
  column(4, "     ", fileInput("upload", "上传文件", accept = c(".txt")))
),
sidebarLayout(
  sidebarPanel(
    actionButton("update", "Change"),
    hr(), # 创建一个代表html标签的R对象
    sliderInput("freq", "词语最小频率:", min = 1,  max = 50, value = 15),
    sliderInput("max", "词语最大重复次数:", min = 1,  max = 300,  value = 100)
  ),

  # Show Word Cloud
  mainPanel(
    plotOutput("plot1")
  )
)
```

以上为`Word cloud 1`部分主体布局的代码实现。

`server` 部分

```
book <- reactive({
    req(input$upload)  # 保证数据上传完才开始运行计算
    ext <- tools::file_ext(input$upload$name)
    switch(ext,
           txt = readLines(input$upload$datapath, encoding = "UTF-8"),
           validate("输入无效! 请上传.txt 文件。")
    )
  })
  terms <- reactive({
    input$update
    isolate({ # 响应阻断器
      withProgress({
        setProgress(message = "Processing corpus...")
        textProcess(book())
      })
    })
  })
  output$plot1 <- renderPlot({
    wordcloud(names(terms()), terms(), scale = c(4,0.5),
                  min.freq = input$freq, max.words = input$max,
                  colors = brewer.pal(8, "Dark2"))
  })
```

```
  output$download1 <- downloadHandler(
    filename = "wordcloud.png",
    content = function(file) {
      png(file)
      wordcloud(names(terms()), terms(), scale = c(4,0.5),
                min.freq = input$freq, max.words = input$max,
                colors = brewer.pal(8, "Dark2"))
      dev.off()
    }
  )
```

`book`为传入的文件，`terms`为运算后的单词-频率矩阵，`isolate`函数作为上传文件后的相应阻断器，需要按下“Begin!”按钮后才会开始画图。

`Word cloud 2`

`Global`部分

```
fontFamily <<-  list("Arial", "Combria")
color <<- list("暗色系" = "random-dark",
               "亮色系" = "random-light")
backgroundColor <<- list("灰色" = "gray",
                         "黑色" = "black")
minRotation <<- list("-PI" = -pi, "-3PI/4" = -3*pi/4,
                     "-PI/2" = pi/2, "-PI/4" = -pi/4)
maxRotation <<- list("PI" = pi, "3PI/4" = 3*pi/4,
                     "PI/2" =  pi/2, "PI/4" = pi/4)
shape <<- list("圆形" = "circle", "心形" = "cardioid", "星形" = "star",
               "钻石形" = "diamond", "三角形" = "triangle", "五边形" = "pentagon")
```

主要作用为控制调节字体、颜色、背景颜色、最小与最大旋转角度、形状标签的选择范围。

`ui`部分

```
sidebarLayout(
  # Sidebar with a slider and selection inputs
  sidebarPanel(
    selectInput("fontFamily", "选择字体:", choices = fontFamily),
    selectInput("color", "选择字体颜色:", choices = color),
    selectInput("backgroundColor", "选择背景颜色:", choices = backgroundColor),
    selectInput("minRotation", "选择字体旋转角度范围的最小值:", choices = minRotation),
    selectInput("maxRotation", "选择字体旋转角度范围的最大值:", choices = maxRotation),
    selectInput("shape", "选择图像形状:", choices = shape),
    sliderInput("size", "字体大小:", min = 1,  max = 10, value = 1),
    sliderInput("fontWeight", "字体粗细:", min = 1,  max = 800,  value = 600),
    sliderInput("rotationRatio", "字体旋转比例:", min = 0,  max = 1,  value = 0.4)
  ),

  # Show Word Cloud
  mainPanel(
    plotOutput("plot2", width='100%', height='400px')
  )
```

```
)
```

以上为`Word cloud 2`部分主体布局的代码实现。

`server`部分

```
output$plot2 <- renderPlot({
    wordcloud2(demoFreq, size = input$size, minSize = 0, gridSize = 0,
               fontFamily = input$fontFamily, fontWeight = input$fontWeight,
               color = input$color, backgroundColor = input$backgroundColor,
               minRotation = input$minRotation, maxRotation = input$maxRotation,
               rotateRatio = input$rotateRatio,
               shape = input$shape, ellipticity = 0.65, widgetsize = NULL)

  })
```

调用了`wordcloud2`函数，实现了绘制出可交互式词云图的功能。以上为`wordcloud2`函数及其所需的各项参数。其中，`wordcloud2`函数的调用需要有一个单词-概率文件，此处我选用了该包自带的的`demoFreq`文件。

用户介绍

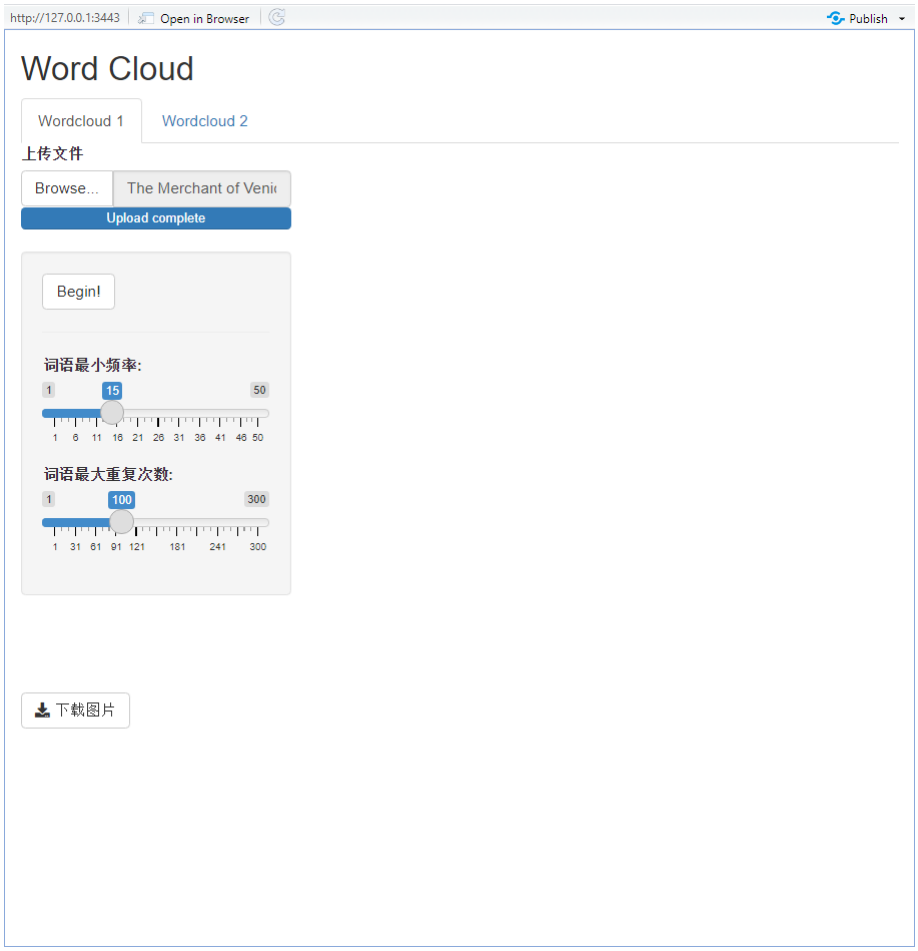
Overview

本网页为一个词云生成网页，接收用户上传的英文txt文件，生成相关单词出现频率的词云图。

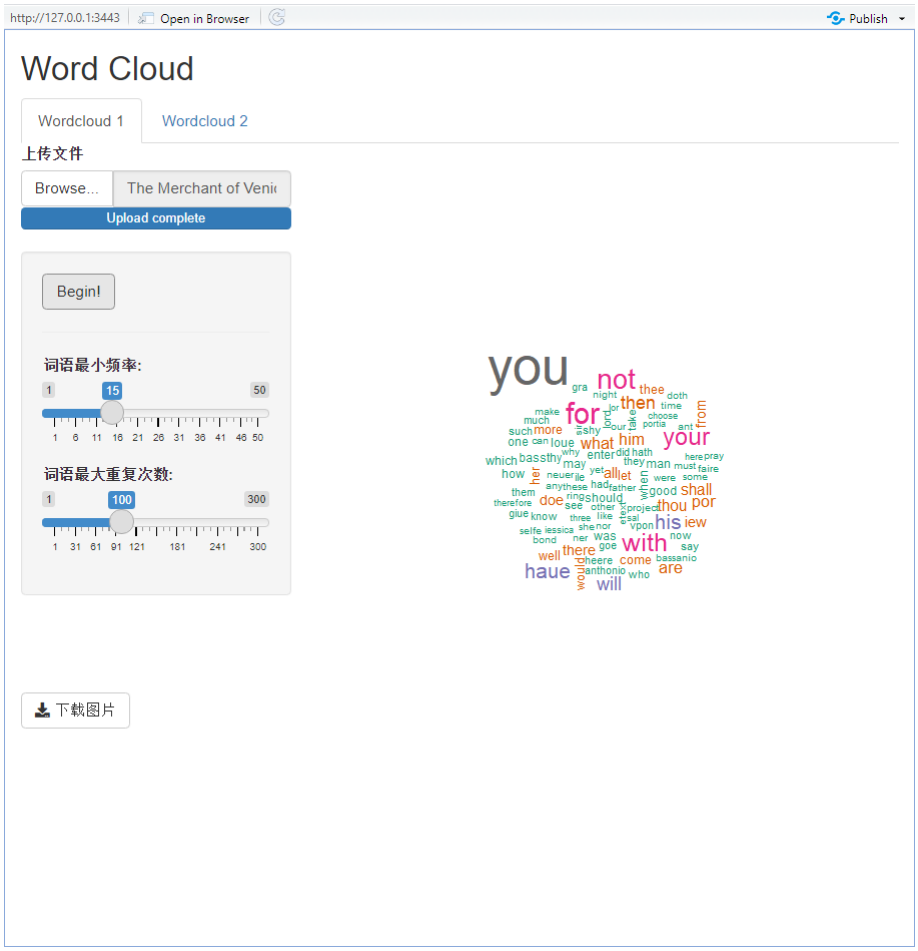
网页界面

Word cloud 1

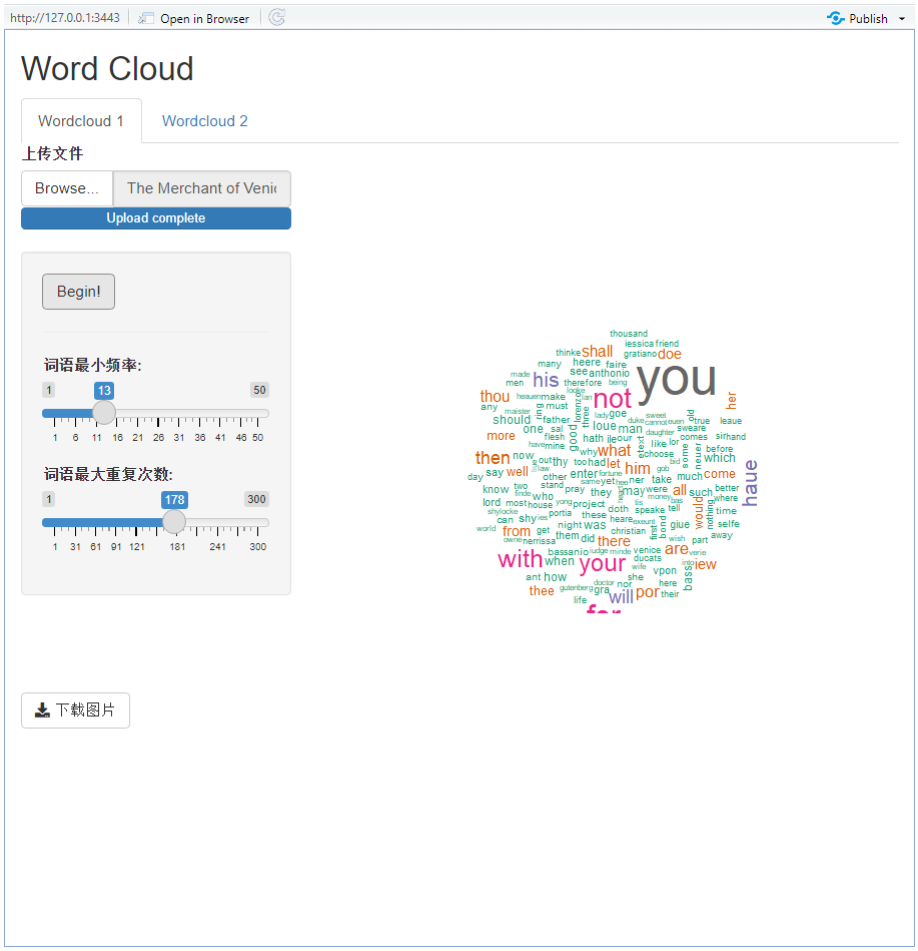
首先，点击“Browse”按钮，上传示例文件”The Merchant of Venice.txt“，如下：



然后单击“Begin!”按钮，开始生成词云图：



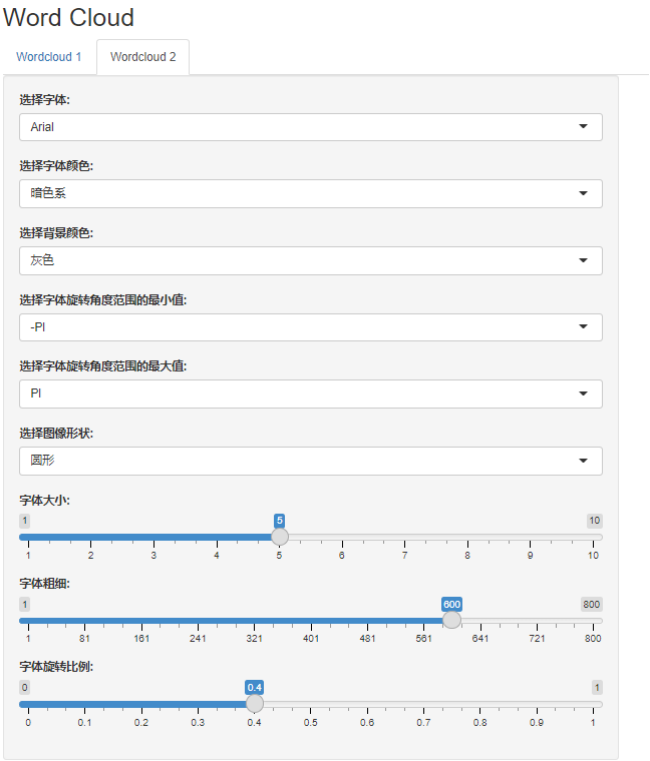
调节拉杆，再次单击“Begin!”按钮，生成新的词云图：



最后，单击“下载图片”按钮，下载格式为“png“的词云图片。

Word cloud 2

点击第二层，进入wordcloud2界面



生成了可交互的词云图，示例如下：