Résumé sur le document <Extraire le contenu d'un pdf avec R>

Jingwen SU

12/20/2020

Introduction

À la fin du cours, tous les étudiants ont téléchargé les documents pertinents sur Github et partagé les résultats avec nous. Afin d'utiliser ces packages d'installation de R de manière plus complète, je vais lire, analyser et évaluer les articles de mes collègues pour améliorer mes lacunes.

Ceci est une introduction à l'article lu dans cet article. Si vous souhaitez en savoir plus, vous pouvez rechercher des références.

- **Title:** Extraire le contenu d'un pdf avec R
- **Auteurs:** Chaymae GASMI
- **Lien sur Github:** Chaymae GASMI

Synthèse du travail en question

Tout d'abord, l'auteur présente le contexte de l'article, expose les raisons de la rédaction de cet article et décrit brièvement le contenu général de l'article. Le texte est divisé en deux parties: "Extraire le contenu d'un fichier PDF en R (deux packages)" et "Nettoyer le résultat afin de pouvoir lancer des analyses sémantiques".

Dans la première partie, l'auteur présente deux packages d'installation: "pdftools" et "tm" en détail dans "Introduction, utilisation et exemples".

Dans la deuxième partie, l'auteur donne des conférences sur l'analyse de texte. Il a présenté en détail les scènes utilisées dans l'analyse de texte et le contenu inclus. Réalisez des exemples pratiques de fonctions associées en termes de fréquence de mot et de nuage de mots.

Contenu principal et explication

1. Le package pdftools

Ce package d'installation est très pratique, il peut nous aider à résoudre le problème de la lecture de PDF dans R. L'introduction et les exemples de l'auteur sont très détaillés.

Pour le contenu PDF importé, nous utiliserons la fonction **pdf_text**.

```
#install.packages(pdftools)
library(pdftools)
download.file("https://www.btboces.org/Downloads/I%20Have%20a%20Dream%20by%20Martin%20Luther%20King%20J
text <- pdf_text("I%20Have%20a%20Dream%20by%20Martin%20Luther%20King%20Jr.pdf")
text1 <- strsplit(text, "\n")
cat(text[1])
```

2.Le package tm

"tm" est plus avancé que le premier. Il a plus d'utilisations.

- **Importation de documents et corpus**

Lors de la comparaison de plusieurs fichiers, nous devons d'abord créer une collection de documents et expliquer le chemin et la méthode de lecture. Pour vérifier les contenus de ces documents on utilise la fonction « inspect »

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

docs <- getwd()
my_corpus <- VCorpus(DirSource(docs, pattern = ".pdf"), readerControl = list(reader = readPDF))

inspect(my_corpus)
writeLines(as.character(my_corpus[[1]]))
```

- **Nettoyage du contenu**

Après avoir obtenu l'article, nous pouvons personnaliser les données. Avant cela, pour protéger les données, nous devons unifier le format. Par exemple, avant de supprimer la ponctuation, nous devons nous assurer qu'il y a des espaces entre la ponctuation et les caractères.

On va se baser sur la fonction **content_transformer** pour créer une fonction **toSpace** qui va nous permettre de mettre un espace entre les ponctuations et le contenu du PDF.

Ensuite, nous pouvons personnaliser les données via des fonctions associées, telles que:

- on peut procéder à la suppression des ponctuations en utilisant la fonction **removePunctuation**
- on va rendre toutes les lettres dans les textes en minuscule avec la fonction **tolower**.
- on va éliminer les nombres avec la fonction **removeNumbers**.

```
toSpace<-content_transformer(function(x,pattern) {return(gsub(pattern," ",x))})

my_corpus<-tm_map(my_corpus, removePunctuation)

my_corpus<- tm_map(my_corpus, content_transformer(tolower))

my_corpus<- tm_map(my_corpus, removeNumbers)
```

3.Analyse des textes

pour savoir la fréquence de chaque mot dans le corpus on utilise la fonction **colSums**.

```
freq<-colSums(as.matrix(dtm))
```

4.Le Wordcloud

Les nuages de mots peuvent nous aider à mieux analyser visuellement les documents. Cela peut être très courant dans les travaux futurs. Il est nécessaire.

a.Installation

Pour générer des nuages de mots on doit télécharger le package **wordcloud** dans R ainsi que le package **RcolorBrewer** pour les couleurs.

```
#install.pakages(wordcloud)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
#install.packages(RColorBrewer)
library(RColorBrewer)
```

b.Création d'une matrice de termes de document

On va créer un nouveau corpus, et on crée un dataframe contenant chaque mot dans la première colonne et leur fréquence dans la deuxième colonne.Cela peut être fait avec la fonction **TermDocumentMatrix** du package tm.

```
article <- Corpus(VectorSource(text))
tm <- TermDocumentMatrix(article)
matrix <- as.matrix(tm)
words <- sort(rowSums(matrix),decreasing=TRUE)
df <- data.frame(word = names(words),freq=words)
```

c. Générer le nuage de mots

```
set.seed(1234)
wordcloud(words = df$word, freq = df$freq, min.freq = 1, max.words=200, random.order=FALSE, rot.per=0.3
```

Evaluation et résumer

Selon mes critères d'évaluation, je pense que cet article est très bon. L'article est très introduction pratique, et très détaillées et des exemples, presque sans aucun problème de mise en forme. Mais je pense que son seul inconvénient est qu'il ne montre pas les résultats de l'opération. Les résultats ne peuvent pas être vus intuitivement.