

R 데이터(텍스트) 마이닝 및 시각화 분석

Seongmin Mun

19th January 2022



아주대학교
AJOU UNIVERSITY

Outline

정규표현식

정규표현식

Dan Jurafsky

- ▶ **Website:** <https://web.stanford.edu/~jurafsky/>
- ▶ **Google Scholar:**
<https://scholar.google.co.il/citations?hl=en&user=uZg9l58AAAAJ>
- ▶ **CS124 (From Languages to Information):**
https://www.youtube.com/channel/UC_48v322owNVtORXuMeRmpA
- ▶ **Regular Expression:**
https://www.youtube.com/watch?v=808M7q8QX0E&list=PLaZQkZp6WhWy4_bClrW9EGQKnUUD9yp8V&index=1
- ▶ **System (regexpal):** <https://www.regexpal.com/>

패키지 설치

```
> install.packages("stringr")
URL 'https://cran.rstudio.com/bin/macosx/contrib/4.1/stringr_1.4.0.tgz'을 시도합니다
Content type 'application/x-gzip' length 210036 bytes (205 KB)
=====
downloaded 205 KB

The downloaded binary packages are in
  /var/folders/f9/tcyhkyhx32q6nxcypnp1rl2c0000gn/T//Rtmp20rPAa/downloaded_packages
> library(stringr)
```

정규 표현식 문법-1

```
> # 1) 정규 표현식 문법
> # .기호 # a와 c 사이에는 어떤 1개의 문자라도 올 수 있다.
> texts <- c("kkk","abc","akc")
> str_detect(texts, "a.c")
[1] FALSE TRUE TRUE
>
>
> # ?기호 # ?앞의 문자가 존재할 수도 있고 존재하지 않을 수도 있는 경우를 나타낸다.
> texts <- c("abbc","abc","ac")
> str_detect(texts, "ab?c")
[1] FALSE TRUE TRUE
```

정규 표현식 문법-2

```
> # *기호 # *은 바로 앞의 문자가 0개 이상일 경우를 나타낸다.  
> texts <- c("a", "ac", "abc", "abbbbc")  
> str_detect(texts, "ab*c")  
[1] FALSE TRUE TRUE TRUE  
>  
>  
> # +기호 # 앞의 문자가 최소 1개 이상이어야 출력한다.  
> texts <- c("ac", "abc", "abbbbc")  
> str_detect(texts, "ab+c")  
[1] FALSE TRUE TRUE
```

정규 표현식 문법-3

```
> # ^기호 #시작되는 문자열을 지정한다.  
> texts <- c("bbc", "zab", "abz", "abzsdfs")  
> str_detect(texts, "^ab")  
[1] FALSE FALSE TRUE TRUE  
>  
>  
>  
> # {숫자} 기호 # 해당 문자를 숫자만큼 반복한 것을 나타낸다.  
> texts <- c("ac", "abb", "abbbbbc", "abbc")  
> str_detect(texts, "ab{2}c")  
[1] FALSE FALSE FALSE TRUE
```


생각하기

- ▶ 단어의 형태가 다른 `woodchuck`를 모두 인식할 수 있는 정규 표현식은 어떻게 작성할까?

정규 표현식 모듈 함수-1

```
> # 2) 정규 표현식 모듈 함수
> strings <- c("R programming is fun.,"Text mining by using R","This special lecture is based on
  R programming which is a useful computer programming language for data analysis.")
>
> # str_detect #검색하고자 하는 문자열이 인풋된 문장에 존재하는지 알려준다.
> str_detect(strings, "is")
[1] TRUE FALSE TRUE
```

정규 표현식 모듈 함수-2

```
> # str_locate # 검색하고자 하는 표현이 어디에 위치하는지 알려준다.  
> str_locate(strings,"is")  
      start end  
[1,]    15  16  
[2,]    NA  NA  
[3,]     3   4  
> str_locate_all(strings,"is")  
[[1]]  
      start end  
[1,]    15  16  
  
[[2]]  
      start end  
  
[[3]]  
      start end  
[1,]     3   4  
[2,]    22  23  
[3,]    54  55  
[4,]   111 112
```

정규 표현식 모듈 함수-3

```
> str_locate_all(strings,"(\\sis\\s)")  
[[1]]  
      start end  
[1,]     14  17  
  
[[2]]  
      start end  
[1,]     21  24  
[2,]     53  56
```

정규 표현식 모듈 함수-4

```
> #str_extract() # 매치 되는 문자열을 보여준다.  
> str_extract(strings,"is")  
[1] "is" NA      "is"  
> str_extract_all(strings,"is")  
[[1]]  
[1] "is"  
  
[[2]]  
character(0)  
  
[[3]]  
[1] "is" "is" "is" "is"  
  
> str_extract_all(strings,"(\\sis\\s)")  
[[1]]  
[1] " is "  
  
[[2]]  
character(0)  
  
[[3]]  
[1] " is " " is "
```

정규 표현식 모듈 함수-5

```
> #str_match() # 일치하는 문자열을 리스트의 형태로 보여준다.  
> str_match(strings,"is")  
      [,1]  
[1,] "is"  
[2,] NA  
[3,] "is"  
> str_match_all(strings,"is")  
[[1]]  
      [,1]  
[1,] "is"  
  
[[2]]  
      [,1]  
  
[[3]]  
      [,1]  
[1,] "is"  
[2,] "is"  
[3,] "is"  
[4,] "is"
```

정규 표현식 모듈 함수-6

```
> #str_split() # 원하는 패턴으로 문자열을 n개로 분할한다.  
> str_split(strings, " ", 2) # 공백을 기준으로 2개의 분할된 문자열 생성  
[[1]]  
[1] "R"                "programming is fun."  
  
[[2]]  
[1] "Text"            "mining by using R"  
  
[[3]]  
[1] "This"  
  
[2] "special lecture is based on R programming which is a useful computer programming language for data analysis."
```

정규 표현식 모듈 함수-7

```
> str_split(strings, "\\s", 3)
[[1]]
[1] "R"          "programming" "is fun."

[[2]]
[1] "Text"      "mining"      "by using R"

[[3]]
[1] "This"

[2] "special"

[3] "lecture is based on R programming which is a useful computer programming language for data a
nalysis."
```


정규 표현식 모듈 함수-8

```
> #str_replace() # 원하는 문자열을 입력한 문자열로 변환한다.
> str_replace(strings, "programming", "language")
[1] "R language is fun."

[2] "Text mining by using R"

[3] "This special lecture is based on R language which is a useful computer programming language
for data analysis."
> str_replace_all(strings, "programming", "language") # 전체 문자열 변환
[1] "R language is fun."

[2] "Text mining by using R"

[3] "This special lecture is based on R language which is a useful computer language language for
data analysis."
```

생각하기

- ▶ **CS124 (From Languages to Information):**
https://www.youtube.com/watch?v=1CSVy9JbbK0list=PLaZQkZp6WhWy4_bClrW9EGQKnUUD9yp8Vindex=2
- ▶ **ELIZA:** <https://web.njit.edu/~ronkowit/eliza.html>