



# 13주차: 텍스트 마이닝

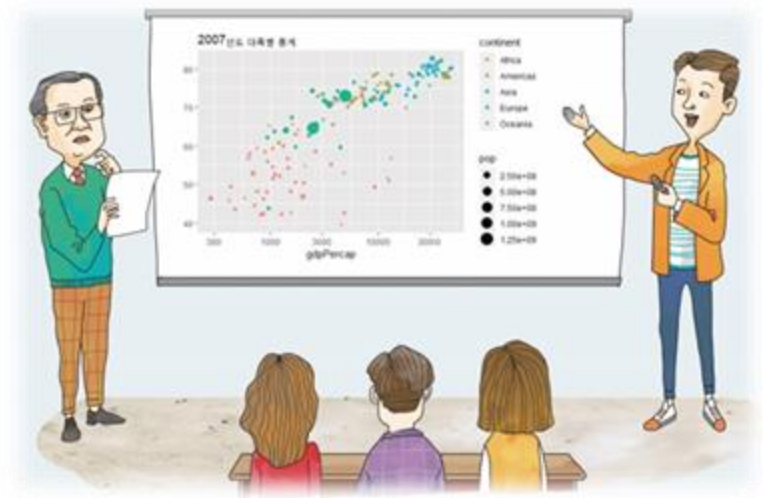
**ChulSoo Park**

School of Computer Engineering & Information Technology  
Korea National University of Transportation

# 11

## CHAPTER

# 텍스트 마이닝



## CONTENTS

11.1 텍스트 마이닝 기초

11.2 DTM 구축

11.3 단어 구름

11.4 문서 분류

**11.5 영어 텍스트 마이닝을 통한 한국어 처리**

**11.6 KoNLP를 이용한 한국어 텍스트 마이닝**

요약

## ■ 한글 DTM 구축 : 예제, 위키피디아의 “빅 데이터” 문서

- [https://ko.wikipedia.org/wiki/%EB%B9%85\\_%EB%8D%B0%EC%9D%B4%ED%84%B0](https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0)



대문  
최근 바뀐  
요즘 화제  
임의의 문서로  
기부

사용자 모임  
사랑방  
사용자 모임  
관리 요청

편집 안내  
도움말  
정책과 지침  
질문방

도구

여기를 가리키는 문서  
가리키는 글의 최근 바뀐  
파일 올리기  
특수 문서 목록  
고유 링크  
문서 정보  
이 문서 인용하기  
위키데이터 항목

인쇄/내보내기  
책 만들기  
PDF로 다운로드  
인쇄중 관

다른 프로젝트

위키미디어 공용

위키백과 검색

문서 토론

읽기 편집 역사 보기

- 과학의 달 에디터톤이 4월 1일부터 30일까지 진행됩니다.
- 2021년 2분기 정비 에디터톤이 4월 16일부터 18일까지 진행됩니다.

## 빅 데이터

위키백과, 우리 모두의 백과사전.

**빅 데이터**(영어: big data)란 기존 **데이터베이스** 관리도구의 능력을 넘어서는 대량(수십 **테라바이트**)의 정형 또는 심지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한<sup>[1]</sup> 데이터로부터 가치를 추출하고 결과를 분석하는 기술<sup>[2]</sup>이다. 즉, 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다.

다양한 종류의 대규모 데이터에 대한 생성, 수집, 분석, 표현을 그 특징으로 하는 빅 데이터 기술의 발전은 다변화된 현대 사회를 더욱 정확하게 예측하여 효율적으로 작동케한다. 개인화된 현대 사회 구성원마다 맞춤형 정보를 제공, 관리, 분석이 가능해 과거에는 불가능했던 기술을 실현시키기도 한다.

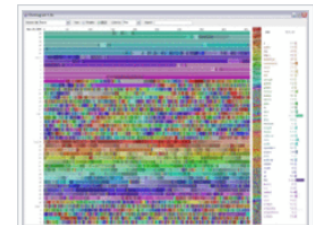
이같이 빅 데이터는 정치, 사회, 경제, 문화, 과학 기술 등 전 영역에 걸쳐서 사회와 인류에게 가치 있는 정보를 제공할 수 있는 가능성을 제시하며 그 중요성이 부각되고 있다.

하지만 빅데이터의 문제점은 바로 사생활 침해와 보안 측면에 자리하고 있다. 빅데이터는 수많은 개인들의 수많은 정보의 집합이다. 그렇기에 빅데이터를 수집, 분석할 때에 개인들의 사적인 정보까지 수집하여 관리하는 빅브라더의 모습이 될 수도 있는 것이다. 그리고 그렇게 모은 데이터가 보안 문제로 유출된다면, 이 역시 거의 모든 사람들의 정보가 유출되는 것이기에 큰 문제가 될 수 있다.

**세계 경제 포럼**은 2012년 떠오르는 10대 기술 중 그 첫 번째를 빅 데이터 기술로 선정<sup>[3]</sup> 했으며 **대한민국 지식경제부** R&D 전략기획단은 IT 10대 핵심기술 가운데 하나로 빅 데이터를 선정<sup>[4]</sup> 하기도 했다.

### 목차 [숨기기]

- 정의
- 특징<sup>[8]</sup>과 의미
  - 빅데이터의 새로운 V<sup>[9]</sup>

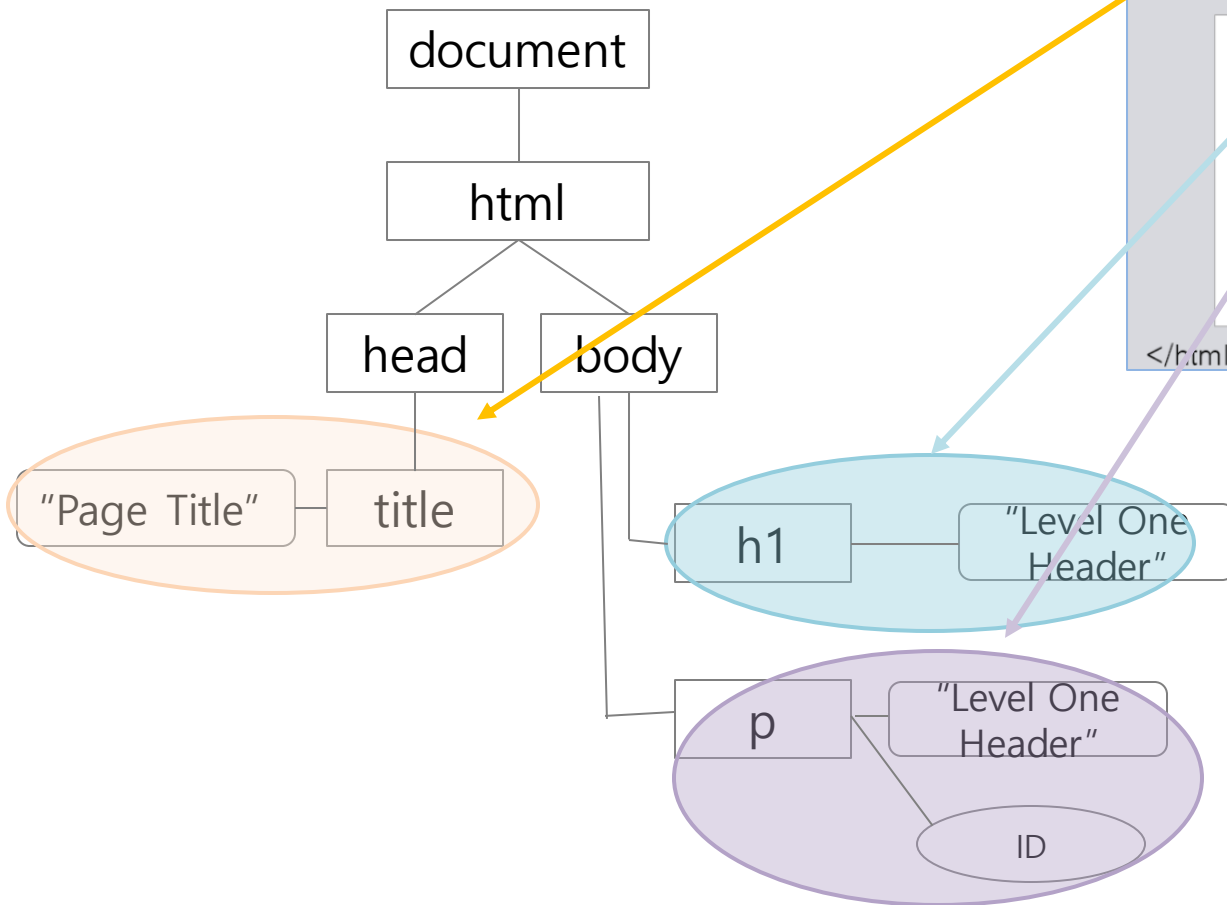


위키백과의 편집 현황의 시각화 자료(BM 작성). 수 **테라바이트**의 용량을 지닌 위키백과의 텍스트 및 이미지 자료는 빅 데이터의 고전적 사례에 속한다.



## html의 구조

DOM tree for simple.html





```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML문서의 제목입니다.</title>
  </head>
  <body>
    <h1>제목 크기1입니다.</h1>
    <h2>제목 크기2입니다.</h2>
    <p>이 부분은 단락입니다.</p>
  </body>
</html>
  
```

The HTML code snippet shows the structure of a simple HTML document. The root element is <html>, which contains <head> and <body> elements. The <head> element contains a <title> element with the text 'HTML문서의 제목입니다.'. The <body> element contains three elements: <h1> (text '제목 크기1입니다.'), <h2> (text '제목 크기2입니다.'), and <p> (text '이 부분은 단락입니다.'). The elements are color-coded: <title> is orange, <h1> is light blue, and <h2> is light blue, and <p> is light purple.

## ■ 예제) 위키의 “빅 데이터” 설명 문서

```
Console C:/RSources/     
> library(tm)  
> library(XML)  
> library(wordcloud2)  
> library(snowballc)  
> library(RCurl)  
> t = read_html('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')  
> #t = readLines('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')  
> d = htmlParse(t, asText = TRUE)  
> clean_doc = xpathApply(d, "//p", xmlValue)
```

1. RCurl 라이브러리 : 웹서버와 접속할 수 있게 도와주는 역할

2. tm(Text Mining) Package

- TermDocumentMatrix 란 함수
- tm\_map 함수 : text data 정제
- VectorSource 함수

## 11.5 영어 텍스트 마이닝을 이용한 한국어 처리

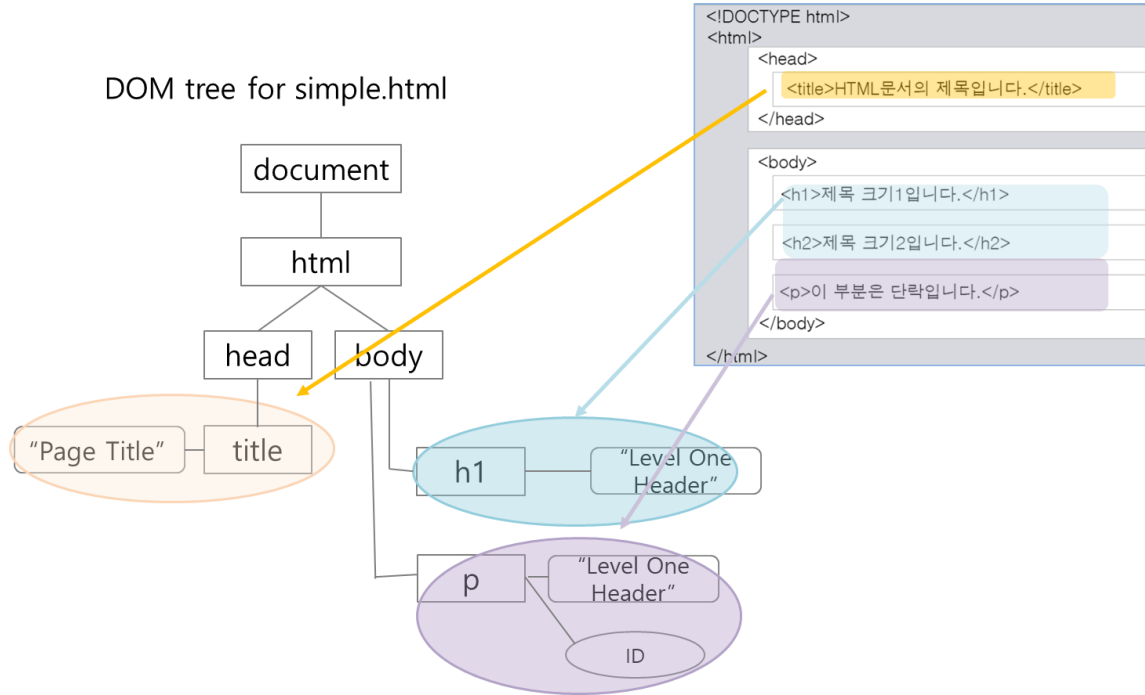
```
Console C:/RSources/ ➤  
> library(tm)  
> library(XML)  
> library(wordcloud2)  
> library(SnowballC)  
> library(RCurl)  
> t = read_html('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')  
> #t = readLines('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')  
> d = htmlParse(t, asText = TRUE)  
> clean_doc = xpathSApply(d, "//p", xmlValue)
```

'빅 데이터'의 UTF-8 표기

3. readlines(read\_html): 라이브러리 : 웹서버와 접속할 수 있게 도와주는 역할
4. htmlParse, xpathSApply 웹 문서를 R의 data형으로 변환
5. SnowballC 라이브러리 : 어간 추출 함수
6. tm\_map : 매개 변수에 따라 텍스트 변환(정제)  
removeNumbers, removeWords 등

# 11.5 영어 텍스트 마이닝을 이용한 한국어 처리

DOM tree for simple.html



C:/RSources/ ↗

```

> t
{html_document}
<html class="client-nojs" lang="ko" dir="ltr">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">\n<meta charset
="UTF-8">\n<title>빅 데이터 - 위키백과, 우리 모두의 백과사전</title>\n<script>document.documentElemen
t.className="client-js";RLCONF={"wgBreakFrames":!1,"wgSeparatorTransformTable" ...
[2] <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject mw-editable page-
빅_데이터 rootpage-빅_데이터 skin-vector action-view skin-vector-search-vue">\n<div class="mw-page-
-container">\n\t<a class="mw-jump-link" href="#content">내용으로 건너뛰기</a>\n\ ...
  
```

## 11.5 영어 텍스트 마이닝을 이용한 한국어 처리

```
> d = htmlParse(t, asText = TRUE)
> clean_doc = xpathSApply(d, "//p", xmlValue)
> str(clean_doc)
> clean_doc[1]
> clean_doc[44]
```

C:/RSources/ 

```
> str(clean_doc)
chr [1:44] "빅 데이터(영어: big data)란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량(수십 테라바이트)의
정형 또는 심지" | __truncated__ ...
> clean_doc[1]
[1] "빅 데이터(영어: big data)란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량(수십 테라바이트)의 정형 또는
심지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한[1] 데이터로부터 가치를 추출하고 결과를 분석하는
기술[2]이다. 즉, 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다.\n"
> clean_doc[44]
[1] "데이비드 캐롤은 2016년 미국 대통령 선거와 2016년 브렉시트 국민투표에 케임브리지 애널리티카(Cambridge Ana1
ytica)가 깊이 관여해 있음을 밝히려고 애쓰면서 영국의 법을 이용해서 케임브리지 애널리티카가 보유하고 있다고 여겨지
는 데이터를 되찾아오려고 노력하고 있다. 그는 런던 소재 고등 법원에 케임브리지 애널리티카와 SCL 선거 캠페인회사(SC
L Elections Ltd)를 언급하며 자신의 데이터를 복구하고 그 출처를 공개하라는 성명을 제출했다. 영국 보수당 국회의원
다미안 콜린스(Damian Noel Thomas Collins MP)가 케임브리지 애널리티카의 대표인 알렉산더 닉스(Alexander Nix)
를 법정에 불러서 심문을 받게 했고, 페이스북의 대표이사 마크 주커버그와 케임브리지 애널리티카의 내부고발자 크리스토
퍼 와일리(Christopher wylie)를 참고인으로 불러 조사가 시작되었다. 데이비드 캐롤은 빅데이터 해킹의 위험에 대해
경고하면서, 대서양 양측에서 규제 압력을 가해서 전세계 기업들이 개인 정보 취급에 대해 보다 투명하게 만들게 해야 한
다고 주장을 계속하고 있다.[45]"
```



## ■ 전처리 수행

Console C:/RSources/ ↗

```
> doc = Corpus(VectorSource(clean_doc))
```

```
> inspect(doc)
```

```
<<SimpleCorpus>>
```

```
Metadata: corpus specific: 1, document level (indexed): 0
```

```
Content: documents: 44
```

[1] 빅 데이터(영어: **big data**)란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량(수십 테라바이트)의 정형 또는 심지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한 [1] 데이터로부터 가치를 추출하고 결과를 분석하는 기술 [2]이다. 즉, 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다.\n

```
> doc = tm_map(doc, content_transformer(tolower))
```

```
> doc = tm_map(doc, removeNumbers)
```

```
> doc = tm_map(doc, removePunctuation)
```

```
> doc = tm_map(doc, stripWhitespace)
```

```
> # -- doc = tm_map(doc, removeWords, stopwords('english'))
```

# 11.5 영어 텍스트 마이닝을 이용한 한국어 처리

```
> doc = tm_map(doc, content_transformer(tolower))  
> doc = tm_map(doc, removeNumbers)  
> doc = tm_map(doc, removePunctuation)  
> doc = tm_map(doc, stripWhitespace)  
> # -- doc = tm_map(doc, removeWords, stopwords('english'))
```

C:/RSources/ ↗

```
> str(clean_doc)  
chr [1:44] "빅 데이터(영어: big data)란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량(수십 테라바이트)의  
정형 또는 심지" | __truncated__ ...  
> clean_doc[1]  
[1] "빅 데이터(영어: big data)란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량(수십 테라바이트)의 정형 또는  
심지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한[1] 데이터로부터 가치를 추출하고 결과를 분석하는  
기술[2]이다. 즉, 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다.\n"  
> clean_doc[44]  
[1] "데이비드 캐롤은 2016년 미국 대통령 선거와 2016년 브렉시트 국민투표에 케임브리지 애널리티카(Cambridge Ana  
lytica)가 깊이 관여해 있음을 밝히려고 애쓰면서 영국의 법을 이용해서 케임브리지 애널리티카가 보유하고 있다고 여겨지  
는 데이터를 되찾  
L Elections Lt  
다미안 콜린스(D  
를 법정에 불러서  
퍼 와일리(Chris  
경고하면서, 대  
다고 주장을 계속
```

```
> inspect(doc)  
<<SimpleCorpus>>  
Metadata: corpus specific: 1, document level (indexed): 0  
Content: documents: 44
```

[1] 빅 데이터영어 big data란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량수십 테라바이트의 정형 또는 심  
지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한 데이터로부터 가치를 추출하고 결과를 분석하는 기술  
이다 즉 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다

[44] 데이비드 캐롤은 년 미국 대통령 선거와 년 브렉시트 국민투표에 케임브리지 애널리티카cambridge analytica가  
깊이 관여해 있음을 밝히려고 애쓰면서 영국의 법을 이용해서 케임브리지 애널리티카가 보유하고 있다고 여겨지는 데이  
터를 되찾아오려고 노력하고 있다 그는 런던 소재 고등 법원에 케임브리지 애널리티카와 sc1 선거 캠페인회사sc1 elec  
tions ltd를 언급하며 자신의 데이터를 복구하고 그 출처를 공개하라는 성명을 제출했다 영국 보수당 국회의원 다미안  
콜린스damian noel thomas collins mp가 케임브리지 애널리티카의 대표인 알렉산더 닉스alexander nix를 법정에  
불러서 심문을 받게 했고 페이스북의 대표이사 마크 주커버그와 케임브리지 애널리티카의 내부고발자 크리스토퍼 와일  
리christopher wylie를 참고인으로 불러 조사가 시작되었다 데이비드 캐롤은 빅데이터 해킹의 위험에 대해 경고하면  
서 대서양 양측에서 규제 압력을 가해서 전세계 기업들이 개인 정보 취급에 대해 보다 투명하게 만들게 해야 한다고 주  
장을 계속하고 있다

## ■ DTM을 구축

Console C:/RSources/ ↗

&gt; dtm = DocumentTermMatrix(doc)

&gt; dim(dtm)

[1] 44 1716

44개 문장 각각을 문서로 간주하여 44개 문서 추출  
이들 문서에서 1716개의 단어를 추출하여 사전 구축

&gt; inspect(dtm)

&lt;&lt;DocumentTermMatrix (documents: 44, terms: 1716)&gt;&gt;

Non-/sparse entries: 2268/73236

Sparsity : 97%

Maximal term length: 24

Weighting : term frequency (tf)

Sample :

'등', '및', '수', '있다', '통해' 등을 단어로 추출하였고,  
'데이터', '데이터를', '데이터의'를 다른 단어로 추출하는 한계

Terms

Docs 년 데이터 데이터를 데이터의 등 및 빅 수 있다 통해

11	0	3	2	1	0	0	1	0	1	0
15	1	0	0	0	2	0	0	1	0	1
16	4	1	1	0	2	2	2	0	0	0
19	2	0	2	3	1	2	1	1	1	1
23	0	0	3	0	2	0	0	1	1	3
36	0	0	0	0	2	2	0	3	2	6
41	2	1	2	0	1	0	1	1	1	0
44	2	0	2	0	0	0	0	0	2	0
7	0	4	4	2	0	0	0	4	4	0
8	2	2	0	3	0	1	1	0	0	1

# 11.5 영어 텍스트 마이닝을 이용한 한국어 처리

## ■ 데이터 프레임으로 변환하고 단어 구름 작성

Console C:/RSources/ ➡

```
> m = as.matrix(dtm)
> v = sort(colsums(m), decreasing = TRUE)
> d = data.frame(word = names(v), freq = v)
> d1 = d[1:500, ] # 500개 단어 표시
> wordcloud2(d1)
```



- ✓ '있다', '통해', '및', '수' 등이 중요 자리 차지
- ✓ '데이터', '빅 데이터', '분석' 이 다른 단어로 간주되어 중요 자리 차지
- ✓ 영어 텍스트 마이닝을 한글에 적용한 한계

## 11.6 KoNLP를 이용한 한글 텍스트 마이닝

- KoNLP는 한국어를 전용으로 처리하는 텍스트 마이닝 라이브러리
  - SystemDic, SejongDic, NIADic이라는 세 종류의 사전을 지원함

### KoNLP package install error 패키지 설치 오류 (2020. 1. 15. 이후) 및 해결방법

R + Textmining(텍스트마이닝) · 2020. 2. 12. 17:00

현재 한글 텍스트 마이닝에서 가장 사랑받는(아니 거의 필수인) "KoNLP" package가 코드 내부적인 문제로 CRAN에서 삭제되었습니다

몇년간 의심없이 써온 package가 삭제되었다는 사실에 적잖이 당황했습니다만..

구글링으로 여러 교수님들의 도움을 받아 간신히 해결하였습니다

## 11.6 KoNLP를 이용한 한글 텍스트 마이닝

### ■ KoNLP는 한국어를 전용으로 처리하는 텍스트 마이닝 라이브러리

#### ■ KoNLP 설치 방법

```
install.packages("multilinguer")  
library(multilinguer)  
install_jdk()  
install.packages(c('stringr', 'hash', 'tau', 'Sejong', 'RSQLite', 'devtools'), type = "binary")  
install.packages("remotes")  
remotes::install_github('haven-jeon/KoNLP', upgrade = "never", INSTALL_opts=c("--no-multiarch"))  
library(KoNLP)
```

## 11.6 KoNLP를 이용한 한글 텍스트 마이닝

- KoNLP는 한국어를 전용으로 처리하는 텍스트 마이닝 라이브러리
  - SystemDic, SejongDic, NIADic이라는 세 종류의 사전을 지원함

```
Console C:/RSources/
> install.packages("KoNLP")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Warning in install.packages :
  package 'KoNLP' is in use and will not be installed
> library(KoNLP)
> useSystemDic()
Backup was just finished!
Error in `[.data.frame`(result_dic, , 2) : undefined columns selected
> useSejongDic()
Backup was just finished!
370957 words dictionary was built.
> useNIADic()
Backup was just finished!
1213109 words dictionary was built.
```

# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## ■ KoNLP는 한국어를 전용으로 처리하는 텍스트 마이닝 라이브러리

### 1. KoNLP 에러 해결 수동설치 -----

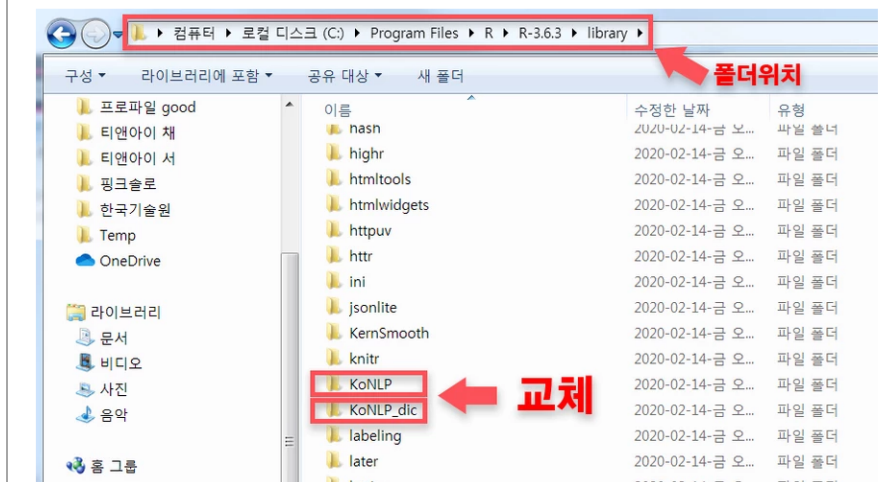
#### 1. 제일먼저 할 일

폴더 위치 --> C:/Program Files/R/R-3.6.3/library <-- 들어가서

\*R-3.6.3 은 설치한 R버전에 맞추어야 합니다.

예: R-3.6.0 설치했다면

폴더 위치 --> C:/Program Files/R/R-3.6.0/library



다운로드한 위 2개를 폴더에 넣습니다. ( 이미 있다면 덮어쓰기 )

다운로드 ▼네이버카페

<https://cafe.naver.com/gachilabs/321>

<https://hhty73.wixsite.com/ihty73/single-post/2020/06/18/konlp-%EC%98%A4%EB%A5%98-%ED%95%B4%EA%B2%B0>



# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## 2. 하단 패키지 설치

#주의 : 위에 2개의 폴더를 넣었기 때문에 KoNLP 는 설치하지 않습니다

이제, 아래 코드를 모두 진행합니다.

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre1.8.0_291')
```

# 코드 마지막 번호 **291** 는 본인 컴퓨터에 설치된 자바의 버전 번호로 맞춰줍니다.

(2021.05.16기준)

```
Sys.getenv("JAVA_HOME")
```

```
install.packages("tau")  
install.packages("rJava")  
install.packages("hash")  
install.packages("hask")  
install.packages("vctrs")  
install.packages("Sejong")  
install.packages("devtools")  
install.packages("RSQLite")
```

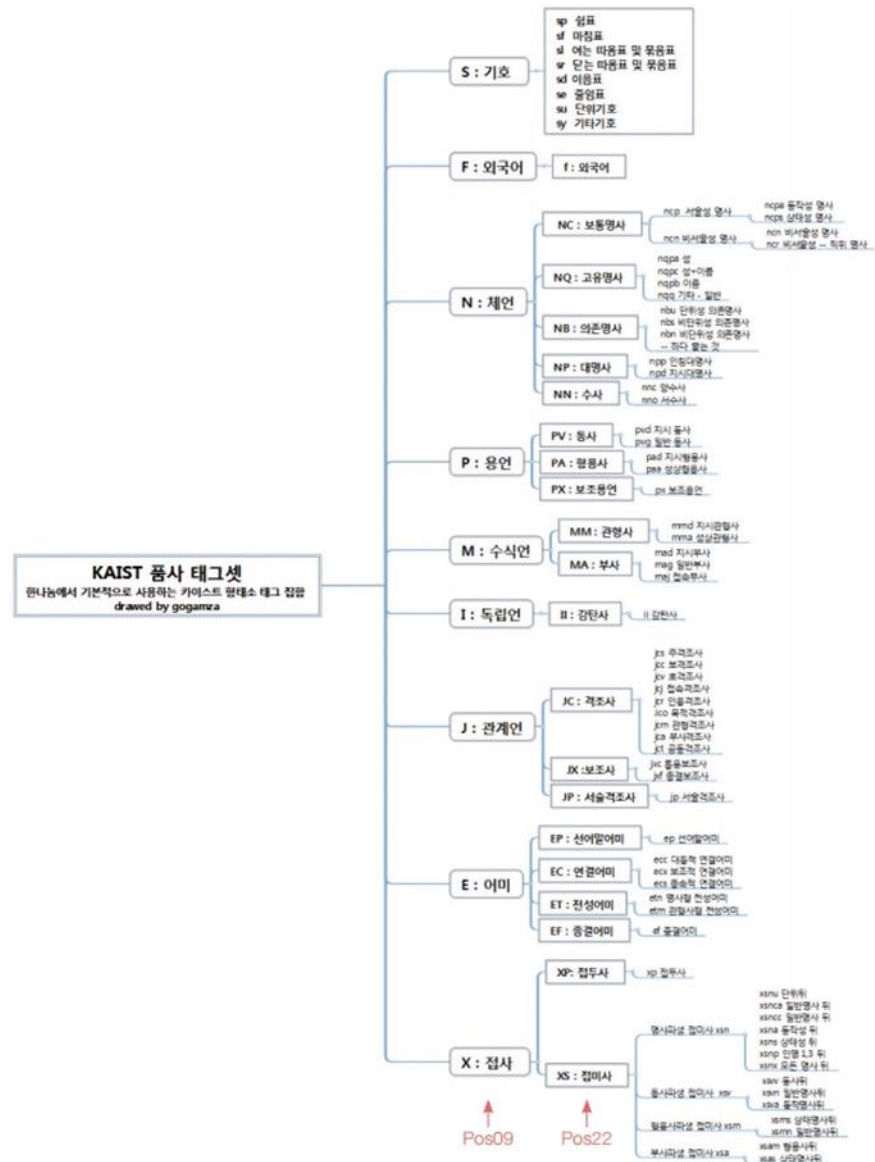
```
library("tau")  
library("rJava")  
library("hash")  
library("hask")  
library("vctrs")  
library("Sejong")  
library("devtools")  
library("RSQLite")
```

```
library("KoNLP")
```

# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## ■ 한국어 품사 태그셋

- Pos09는 9 종류로 품사 구분
- Pos22는 22 종류로 품사 구분



# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## KAIST 품사 태그셋

기호(s)		1. sp(업표)                      2. sf(마침표) 3. sl(여는 따옴표 및 묶음표) 4. sr(닫는 따옴표 및 묶음표) 5. sd(이음표)                      6. se(줄임표) 7. su(단위 기호)                      8. sy(기타 기호)
외국어(f)		9. f(외국어)
배언(n)	보통 명사(nc)	
	서술성 명사(ncp)	10. ncpa(동작성 명사)                      11. ncps(상태성 명사)
	비서술성 명사(ncn)	12. ncn(비서술성 명사)
	고유명사(nq)	13. nq(고유명사)
	의존명사(nb)	14. nbu(단위성 의존 명사)                      15. nbn(비단위성 의존 명사)
	대명사(np)	16. npp(인칭 대명사)                      17. npd(지시 대명사)
용언(p)	수사(nn)	18. nnc(양수사)                      19. nno(서수사)
	동사(pv)	20. 일반 동사(pvg)                      21. 지시 동사(pvd)
	형용사(pa)	22. 성상 형용사(paa)                      23. 지시 형용사(pad)
	보조 용언(px)	24. 보조용언(px)
수식언(m)	관형사(mm)	25. 성상 관형사(mma)                      26. 지시 관형사(mmd)
	부사(ma)	27. 일반 부사(mag)                      28. 지시 부사(mad)
		29. 접속 부사(maj)
독립언(i)	감탄사(ii)	30. 감탄사(ii)
관계언(j)	격조사(jc)	31. 주격 조사(jcs)                      32. 목적격 조사(jco) 33. 보격 조사(jcc)                      34. 관형격 조사(jcm) 35. 호격 조사(jcv)                      36. 부사격 조사(jca) 37. 접속격 조사(jcj)                      38. 공동격 조사(jct) 39. 인용격 조사(jcr)
	서술격 조사(jp)	40. 서술격 조사(jp)
	보조사(jx)	41. 통용 보조사(jxc)                      42. 종결 보조사(jxf)
어미(e)	종결 어미(ef)	43. 종결 어미(ef)
	선어말 어미(ep)	44. 선어말 어미(ep)
	연결 어미(ec)	45. 대동작 연결 어미(ecc)                      46. 종속작 연결 어미(ecs) 47. 보조적 연결 어미(ecx)
	전설 어미(et)	48. 명사형 어미(etn)                      49. 관형사형 어미(etm)
접사(x)	접두사(xp)	50. 접두사(xp)
	접미사(xs)	51. 명사 파생 접미사(xsn)                      52. 동사 파생 접미사(xsv) 53. 형용사 파생 접미사(xsm)                      54. 부사 파생 접미사(xsa)

↑ Pos09  
↑ Pos22

# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## ■ 한글 형태소 분석 예제

- 형태소(morpheme)란 언어를 구성하는 가장 작은 문법 요소
- 형태소 분석이란 문장을 형태소 단위로 분할하는 작업

'너에게'를 '너'라는 대명사(NP)와 '에게'라는 격조사(JC)로 분해

```
> useSejongDic()
Backup was just finished!
370957 words dictionary was built.
> s='너에게 묻는다 연탄재 함부로 발로 차지 마라 너는 누구에게 한번이라도 뜨거운 사람이었느냐'
> extractNoun(s)
[1] "너" "연탄재" "발" "차" "너"
[6] "누구" "한" "번" "사람이었느" "냐"
> simplePos22(s)
$너에게
[1] "너/NP+에게/JC"
$묻는다
[1] "문/PV+는다/EF"
$연탄재
[1] "연탄재/NC"
$함부로
[1] "함부로/MA"
$발로
[1] "발/NC+로/JC"
$차지
[1] "차/NC+이/JP+지/EC"
$마라
[1] "마르/PV+아/EC"
$너는
[1] "너/NP+는/JX"
$누구에게
[1] "누구/NP+에게/JC"
$한번이라도
[1] "한/NN+번/NB+이라도/JX"
$뜨거운
[1] "뜨겁/PA+은/ET"
$사람이었느
[1] "사람이었느/NC"
$냐
[1] "냐/NC"
```

extractNoun 함수는 명사를 추출

simplePos22 함수는 Pos22 단계까지 형태소 분석을 수행

# 11.6 KoNLP를 이용한 한글 텍스트 마이닝

## ■ 위키 “빅 데이터” 예제 (11.5절 참조)

```
Console C:/Rsources/
> t = read_html('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')
> d=htmlParse(t, asText=TRUE)
> clean_doc=xpathSapply(d, "//p", xmlValue)
> useSejongDic()
Backup was just finished!
370957 words dictionary was built.
> nouns = extractNoun(clean_doc)
> mnous = unlist(nouns)
> mnous_freq = table(mnous)
> v = sort(mnous_freq, decreasing = TRUE)
> wordcloud2(v) # 모든 단어 표시
> v1 = v[1:200] # 상위 200개 표시
> wordcloud2(v1)
```

```
> inspect(doc)
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 44
```

[1] 빅 데이터 영어 **big data**란 기존 데이터베이스 관리도구의 능력을 넘어서는 대량수집 테라바이트의 정형 또는 심지어 데이터베이스 형태가 아닌 비정형의 데이터 집합조차 포함한 데이터로부터 가치를 추출하고 결과를 분석하는 기술이다 즉 기존의 데이터 베이스로는 처리하기 어려울 정도로 방대한 양의 데이터를 의미한다

```
> extractNoun(clean_doc)
[[1]]
[1] "빅" "데이터(영어:" "big" "data" "기존"
[6] "데이터베이스" "관리" "도구" "능력" "대량(수집"
[11] "테라바이트)" "정형" "데이터베이스" "형태" "비정"
[16] "형" "데이터" "집합" "포함한[1]" "데이터"
[21] "가치" "추출" "결과" "분석" "기술[2]이다"
[26] "즉" "기존" "데이터" "베이스" "처리"
```

- 영어 텍스트 마이닝을 활용한 [그림 11-12]에 비해 큰 향상
- 하지만 여전히 한계 (KoNLP의 한계)
  - ‘빅데이터’와 ‘빅데이터를’이 함께 나타나는 현상
  - ‘한’이 중요한 단어로 등장 (‘포함한’, ‘다양한’ 등에서 ‘한’이 명사로 잘못 추출됨)



## 11.6 KoNLP를 이용한 한글 텍스트 마이닝

```

184 # --- 한글 예외처리 1 ---
185 t2 = read_html('https://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0')
186 d2 = htmlParse(t2, asText = TRUE)
187 clean_doc2 = xpathSApply(d2, "//p", xmlValue)
188
189 doc2 = Corpus(VectorSource(clean_doc2))
190
191 doc2 = tm_map(doc2, removePunctuation)
192 doc2 = tm_map(doc2, removeNumbers)
193 doc2 = tm_map(doc2, stripWhitespace)
194 mystopword = c(stopwords('english'), "한", "것", "등", "수", "및", "있는", "있다", "년")
195 mystopword
196 doc2 = tm_map(doc2, removeWords, mystopword)
197
198 dtm2 = DocumentTermMatrix(doc2)
199 m = as.matrix(dtm2)
200 v = sort(colSums(m), decreasing = TRUE)
201 d = data.frame(word = names(v), freq = v)
202 d1 = d[1:200, ] # 200개 단어 표시
203 wordcloud2(d1)

```





## 11.6 KoNLP를 이용한 한글 텍스트 마이닝

```
mystopword = c(stopwords('english'), "한", "것", "등", "수", "및", "있는", "있다", "년")
mystopword
doc2 = tm_map(doc2, removeWords, mystopword)
```

### 불용어 등록 사용 방법

Console

Jobs x

C:/RSources/ ↗

[85]	"hasn't"	"haven't"	"hadn't"	"doesn't"	"don't"	"didn't"
[91]	"won't"	"wouldn't"	"shan't"	"shouldn't"	"can't"	"cannot"
[97]	"couldn't"	"mustn't"	"let's"	"that's"	"who's"	"what's"
[103]	"here's"	"there's"	"when's"	"where's"	"why's"	"how's"
[109]	"a"	"an"	"the"	"and"	"but"	"if"
[115]	"or"	"because"	"as"	"until"	"while"	"of"
[121]	"at"	"by"	"for"	"with"	"about"	"against"
[127]	"between"	"into"	"through"	"during"	"before"	"after"
[133]	"above"	"below"	"to"	"from"	"up"	"down"
[139]	"in"	"out"	"on"	"off"	"over"	"under"
[145]	"again"	"further"	"then"	"once"	"here"	"there"
[151]	"when"	"where"	"why"	"how"	"all"	"any"
[157]	"both"	"each"	"few"	"more"	"most"	"other"
[163]	"some"	"such"	"no"	"nor"	"not"	"only"
[169]	"own"	"same"	"so"	"than"	"too"	"very"
[175]	"한"	"것"	"등"	"수"	"및"	"있는"
[181]	"있다"	"년"				



# Thank you

