

파이썬을 활용한 실전 웹크롤링

크롤링 데이터 추출 및 전처리

2018.09.01

신준호

jjjjoonno@gmail.com

목차 - 실습

1. Pandas 패키지와 DataFrame

2. cKonlpy와 matplotlib 패키지 학습

3. Gensim 패키지로 텍스트 주제 뽑아내기

Pandas 패키지 학습의 목적.

이제까지는 크롤링한 결과를 List 혹은 Dictionary의 형태로만 확인했습니다. 이제는 실제 업무에 활용하기 위해, Excel 파일로 뽑는 실습을 진행합니다.

Pandas DataFrame

Pandas는 **매트릭스**, **테이블**과 같은 데이터를 처리하기 위해 사용하는 패키지입니다.

이 판다스의 모듈중에 **DataFrame**이라는 모듈은 **Index**와 **column**으로 데이터를 접근 할 수 있는 테이블을 만들어서 다루는데 사용됩니다.
이 DataFrame을 만들고 다루는 방법에 대해 알아보겠습니다.

DataFrame 만들기

Dataframe은 **길이가 같은 리스트 여러개**로 만들 수 있습니다.

```
Dt = Dataframe({'칼럼이름 첫번째' : 첫번째 들어갈 리스트, ...})
```

우선 **문자열로 행이름**을 정해주고, 그 **행에 들어갈 데이터를 리스트 변수**로 넣습니다.

그리고 **심표로 구분하여 여러개의 리스트를 넣어서 데이터 프레임**을 완성합니다.

기존의 Excel(csv)파일을 DataFrame으로 읽기

Pandas는 기존의 파일을 DataFrame으로 읽을 수 있습니다.

```
Dt = read_csv('경로와 파일이름까지', header = '칼럼명의 유무', encoding = '파일의 인코딩')
```

```
Dt = read_excel('경로와 파일이름까지', header = '칼럼명의 유무', encoding = '파일의 인코딩')
```

우선 read 함수의 첫 인자는 파일명입니다. 파일의 경로까지 넣어줘야 한다는 점을 항상 기억해 주시면 됩니다.

그리고 엑셀파일에서 첫 줄을 칼럼명으로 쓸 것인지를 정해주는 header와 파일의 인코딩을 지정해주는 encoding도 항상 맞는 방식으로 채워주면 됩니다.

DataFrame 실제로 만들어 보기

A = [a,a,b,b]

B = [1,2,3,4]

C = [가,나,다,라]

Dt = DataFrame({'에이' : A,
 '비' : B,
 '씨' : C})



	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라

DataFrame에 접근하기

인덱스 뽑기 = `dt.index`

-> `[0,1,2,3]`

칼럼명 모두 뽑기 = `dt.column`

-> `['에이','비','씨']`

칼럼 하나씩 뽑기 = `dt['에이']`

-> `['a','a','b','b']`

앞 5개 보기 = `dt.head()`

-> 앞의 5줄

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라

DataFrame에 접근하기


DataFrame 안의 데이터에 접근하기

`dt.iloc[열번호, 행번호]`

`dt.iloc[2,1] = 3`입니다.

`dt.iloc[:2,1:]`

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



DataFrame에 접근하기

DataFrame 안의 데이터에 접근하기

`dt.loc[조건, 혹은 행 인덱스의 이름]`

`dt.loc[dt['비'] > 2,:]`

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라

DataFrame에 적용되는 함수들(중복제거하기)

`dt.drop_duplicates('에이')`



	에이	비	씨
0	a	1	가
2	b	3	다

-> 에이 열을 기준으로 중복된것이 있으면 제거하는 함수
기본적으로 맨 처음 것만 남기고 밑의 중복을 지워줌.

주의할 점은 dt_dr과 같이 새로운 DataFrame 변수에 정의를 해야 중복제거가 된 상태가 저장이 된다는 점입니다!

DataFrame에 적용되는 함수들(그룹화)

`dt.groupby(['에이']).count()`

-> 데이터 프레임에 **에이**를 기준으로 그룹화 하여 뽑고 개수를 제공합니다.

이때 기준이 되는 칼럼은 **index**의 자리로 들어가게 됩니다.

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



`dt.groupby(['에이']).count()`

에이	비	씨
a	2	2
b	2	2

DataFrame에 적용되는 함수들(그룹화)

`dt.groupby(['에이']).mean()`

-> 데이터 프레임에 **에이**를 기준으로 그룹화 하여 뽑고 수치의 평균을 계산합니다.
이때 문자형이 들어간 칼럼은 자동적으로 제외하고 수치형의 칼럼만 계산합니다.

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



`dt.groupby(['에이']).mean()`

에이	비
a	1.5
b	3.5

DataFrame에 적용되는 함수들(Pivot)

`dt.pivot('비','씨','에이')`

-> 데이터 프레임의 열 행 내용으로 피버팅합니다.

이때 열 행 내용 순으로 칼럼이름이 들어갑니다.

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



`dt.pivot('에이')`

에이	가	나	다	라
1	a	NA	NA	NA
2	NA	a	NA	NA
3	NA	NA	b	NA
4	NA	NA	NA	b

DataFrame에 적용되는 함수들(NA 값 대체하기)

```
dt.pivot('비','씨','에이').nafill('0')
```

-> 데이터 프레임을 **에이**를 기준으로 그룹화 하여 뽑습니다.

이때 기준이 되는 칼럼은 **index**의 자리로 들어가게 됩니다.

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



```
dt.pivot('비','씨','에이').nafill(0)
```

에이	가	나	다	라
1	a	0	0	0
2	0	a	0	0
3	0	0	b	0
4	0	0	0	b

DataFrame에 적용되는 함수들 (concat)

`concat([dt,dt],axis = 0)`

-> 데이터프레임을 위아래로 합칩니다.

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라



`concat([dt,dt],axis = 0)`

	에이	비	씨
0	a	1	가
1	a	2	나
2	b	3	다
3	b	4	라
4	a	1	가
5	a	2	나
6	b	3	다
7	b	4	라

DataFrame에 적용되는 함수들 (concat)

`concat([dt,dt],axis = 1)`

-> 데이터프레임을 양옆으로 합칩니다.

	에이	비	씨	에이	비	씨
0	a	1	가	a	1	가
1	a	2	나	a	2	나
2	b	3	다	b	3	다
3	b	4	라	b	4	라

DataFrame에 적용되는 함수들(merge)

`merge(dt,dt,on = '비' -> Key 칼럼들)`

-> 데이터프레임을 Key를 기준으로 합칩니다.

	씨_x	에이_x	비	에이_y	씨_y
0	가	a	1	a	가
1	나	a	2	a	나
2	다	b	3	b	다
3	라	b	4	b	라

DataFrame에 적용되는 함수들(sort_values)

`dt.sort_values('비', ascending = False)`

-> 비 칼럼을 기준으로 내림차순으로 dt를 정렬합니다.

	에이	비	씨
3	b	4	라
2	b	3	다
1	a	2	나
0	a	1	가

DataFrame 저장하기

`dt.to_excel('저장경로', sheet_name='시트이름', header=맨 윗줄이 칼럼명이라면 True, 아니라면 False(이때는 자동적으로 칼럼명을 지정해줌.))`

-> 데이터프레임을 엑셀파일로 뽑아냅니다.

경로 끝에는 항상 '**파일이름.xlsx**'를 적어 주셔야 합니다.

`to_csv()` 또한 시트이름을 지정하는 명령어를 제외하고 같은 방법으로 사용합니다.

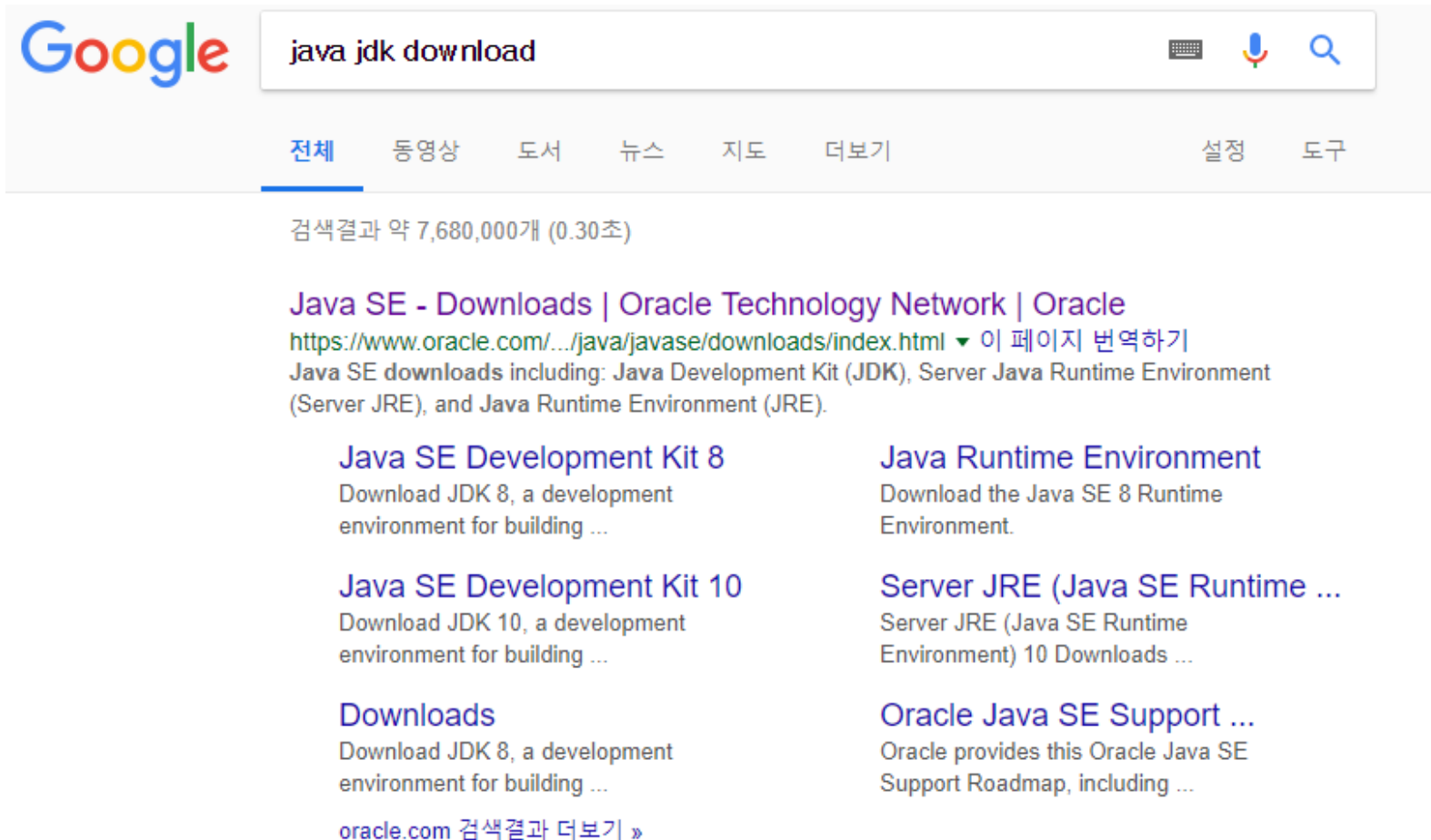
그러나 csv는 텍스트 파일과 비슷하게 `encoding`을 지정해 주어야 합니다.

Python은 기본적으로 utf-8을 사용하기 때문에 그냥 저장을 하게 되면 엑셀에서 열때 한글이 호환되지 않습니다. 그래서 저장할 때는 항상 **`encoding = 'euc-kr' 혹은 'cp-949'`**를 넣어 주어야 합니다.

cKonlpy와 Gensim 패키지 학습의 목적.

두번째 시간에는 텍스트 데이터의 전처리와 시각화를 위한 패키지 두개를 학습합니다. 한글 전처리는 원하는 도메인에 따라 고유명사를 지정해 줘야하는 경우가 많기 때문에 사전을 수정할 수 있는 Customized Konlpy를 사용합니다.

ckonlpy 설치를 위한 JAVA 설치



Google

java jdk download

전체 동영상 도서 뉴스 지도 더보기

설정 도구

검색결과 약 7,680,000개 (0.30초)

Java SE - Downloads | Oracle Technology Network | Oracle
<https://www.oracle.com/.../java/javase/downloads/index.html> ▼ 이 페이지 번역하기
Java SE downloads including: Java Development Kit (JDK), Server Java Runtime Environment (Server JRE), and Java Runtime Environment (JRE).

Java SE Development Kit 8 Download JDK 8, a development environment for building ...	Java Runtime Environment Download the Java SE 8 Runtime Environment.
Java SE Development Kit 10 Download JDK 10, a development environment for building ...	Server JRE (Java SE Runtime ...) Server JRE (Java SE Runtime Environment) 10 Downloads ...
Downloads Download JDK 8, a development environment for building ... oracle.com 검색결과 더보기 »	Oracle Java SE Support ... Oracle provides this Oracle Java SE Support Roadmap, including ...

ckonlpy 설치를 위한 JAVA 설치

Java SE 8u181

Java SE 8u181 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.

[Learn more](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
 - [JDK ReadMe](#)
 - [JRE ReadMe](#)

JDK

[DOWNLOAD](#)

Server JRE

[DOWNLOAD](#)

JRE

[DOWNLOAD](#)

Java SE 7, and Java SE 6 updates

Updates for Java SE 7 released after April 2015, and updates for Java SE 6 released after April 2013 are only available to Oracle Customers through [My Oracle Support](#) (requires support login).

[Java SE Subscription](#) offers users commercial features, access to critical bug fixes, security fixes, and general maintenance.

Early Access Releases

Early access versions of future releases of the JDK and the JRE are available for testing. These early access releases include future update and future major releases. These releases are licensed only for testing, not for use in production.

[DOWNLOAD](#)

JDK 8 Demos and Samples

Demos and samples of common tasks and new functionality available on JDK 8. JavaFX 8 demos and samples are included in the JDK 8 Demos and Samples packages. The source code provided with demos and samples for the JDK is meant to illustrate the usage of a given feature or technique and has been deliberately simplified.

[DOWNLOAD](#)

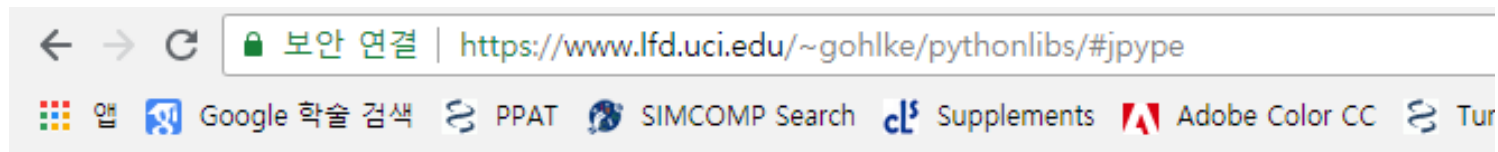
Java SE Development Kit 8u181

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.95 MB	jdk-8u181-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.89 MB	jdk-8u181-linux-arm64-vfp-hflt.tar.gz
Linux x86	165.06 MB	jdk-8u181-linux-i586.rpm
Linux x86	179.87 MB	jdk-8u181-linux-i586.tar.gz
Linux x64	162.15 MB	jdk-8u181-linux-x64.rpm
Linux x64	177.05 MB	jdk-8u181-linux-x64.tar.gz
Mac OS X x64	242.83 MB	jdk-8u181-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.17 MB	jdk-8u181-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.34 MB	jdk-8u181-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.83 MB	jdk-8u181-solaris-x64.tar.Z
Solaris x64	92.11 MB	jdk-8u181-solaris-x64.tar.gz
Windows x86	194.41 MB	jdk-8u181-windows-i586.exe
Windows x64	202.73 MB	jdk-8u181-windows-x64.exe

ckonlpy 설치를 위한 JAVA 설치



JPype allows full access to Java class libraries.

[JPype1-0.6.3-cp27-cp27m-win32.whl](#)

[JPype1-0.6.3-cp27-cp27m-win amd64.whl](#)

[JPype1-0.6.3-cp34-cp34m-win32.whl](#)

[JPype1-0.6.3-cp34-cp34m-win amd64.whl](#)

[JPype1-0.6.3-cp35-cp35m-win32.whl](#)

[JPype1-0.6.3-cp35-cp35m-win amd64.whl](#)

[JPype1-0.6.3-cp36-cp36m-win32.whl](#)

[JPype1-0.6.3-cp36-cp36m-win amd64.whl](#)

[JPype1-0.6.3-cp37-cp37m-win32.whl](#)

[JPype1-0.6.3-cp37-cp37m-win amd64.whl](#)

...

ckonlpy 설치를 위한 JAVA 설치

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.16299.64]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\JunHee>javac -version
javac 1.8.0_181

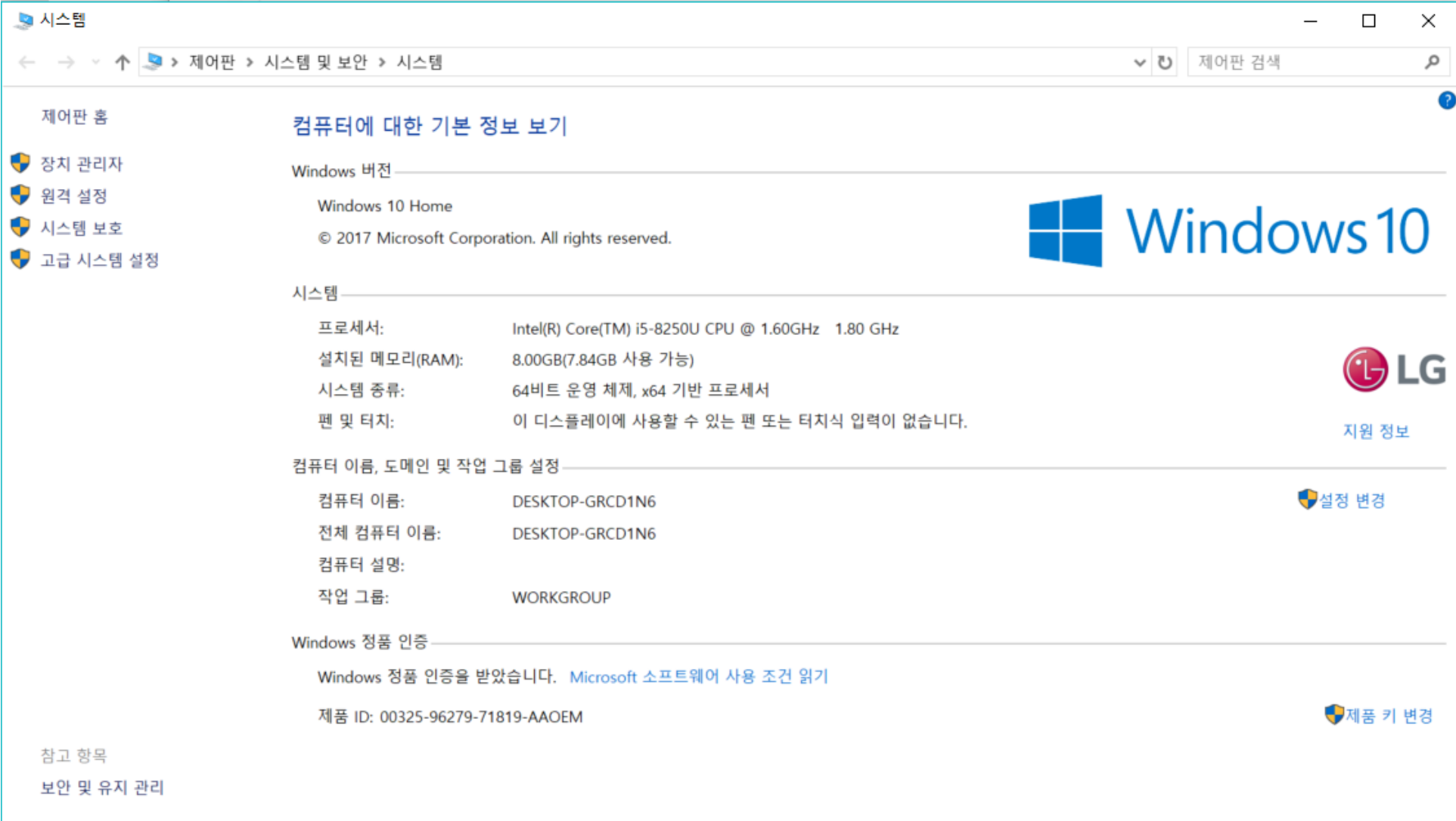
C:\Users\JunHee>javac

Usage: javac <options> <source files>

where possible options include:

-g	Generate all debugging info
-g:none	Generate no debugging info
-g:{lines,vars,source}	Generate only some debugging info
-nowarn	Generate no warnings
-verbose	Output messages about what the compiler is doing
-deprecation	Output source locations where deprecated APIs are used
-classpath <path>	Specify where to find user class files and annotation processors
-cp <path>	Specify where to find user class files and annotation processors
-sourcepath <path>	Specify where to find input source files
-bootclasspath <path>	Override location of bootstrap class files
-extdirs <dirs>	Override location of installed extensions
-endorseddirs <dirs>	Override location of endorsed standards path
-proc:{none,only}	Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]	Names of the annotation processors to run; bypasses default discovery process
-processorpath <path>	Specify where to find annotation processors
-parameters	Generate metadata for reflection on method parameters
-d <directory>	Specify where to place generated class files

ckonlpy 설치를 위한 JAVA 설치

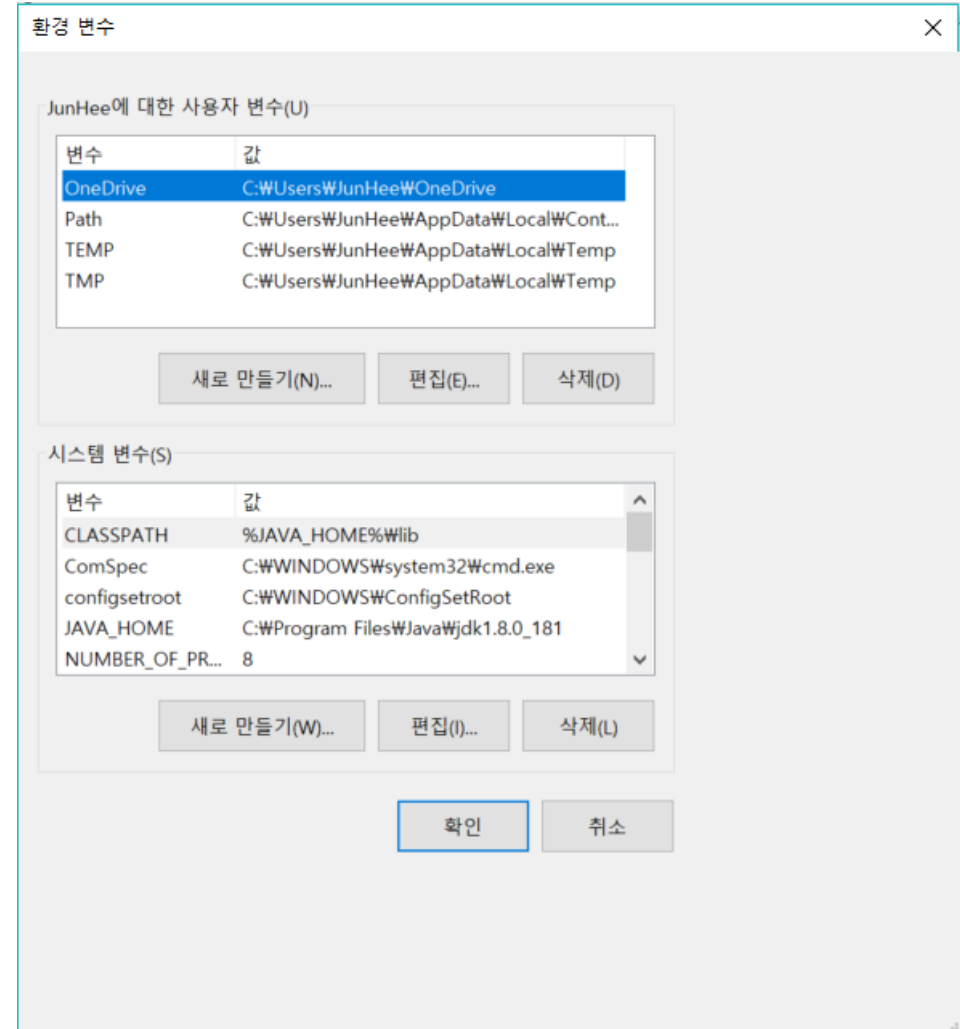
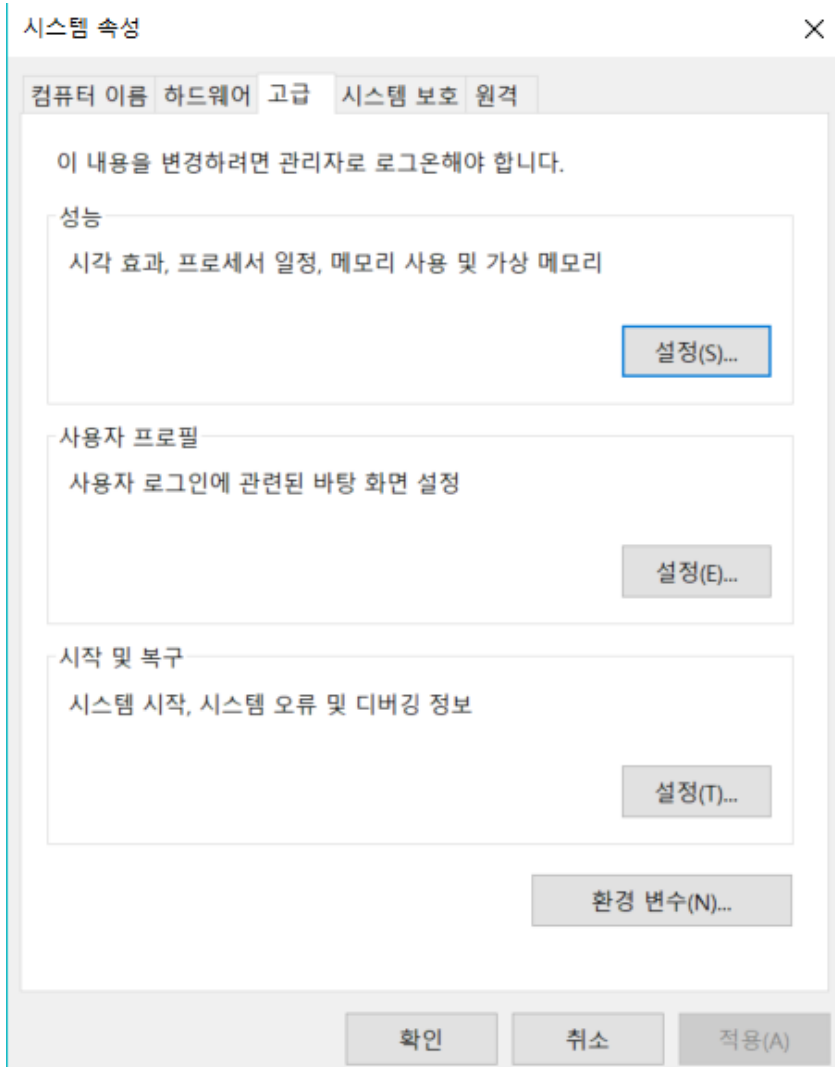


The screenshot shows the Windows 10 'System' window. The left sidebar contains links to 'Control Panel Home', 'Device Manager', 'RemoteApp Settings', 'System Protection', and 'Advanced System Settings'. The main content area is titled 'Computer Basic Information' and displays the following details:

- Windows 버전**: Windows 10 Home, © 2017 Microsoft Corporation. All rights reserved.
- 시스템**:
 - 프로세서: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
 - 설치된 메모리(RAM): 8.00GB(7.84GB 사용 가능)
 - 시스템 종류: 64비트 운영 체제, x64 기반 프로세서
 - 팬 및 터치: 이 디스플레이에 사용할 수 있는 팬 또는 터치식 입력이 없습니다.
- 컴퓨터 이름, 도메인 및 작업 그룹 설정**:
 - 컴퓨터 이름: DESKTOP-GRCD1N6
 - 전체 컴퓨터 이름: DESKTOP-GRCD1N6
 - 컴퓨터 설명:
 - 작업 그룹: WORKGROUP
- Windows 정품 인증**:
 - Windows 정품 인증을 받았습니다. [Microsoft 소프트웨어 사용 조건 읽기](#)
 - 제품 ID: 00325-96279-71819-AAOEM

Additional UI elements include the Windows 10 logo, an LG logo with a 'Support Information' link, and buttons for 'Change Settings' and 'Change Product Key'.

ckonlpy 설치를 위한 JAVA 설치

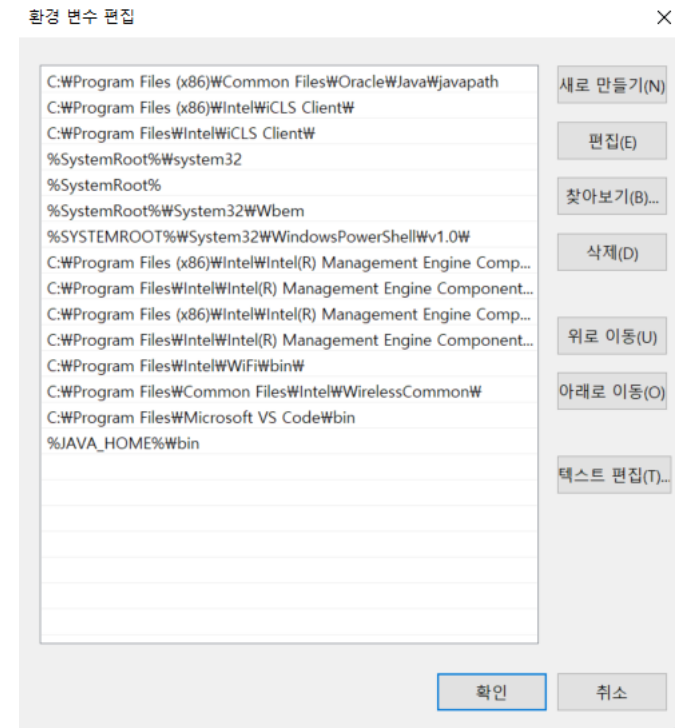
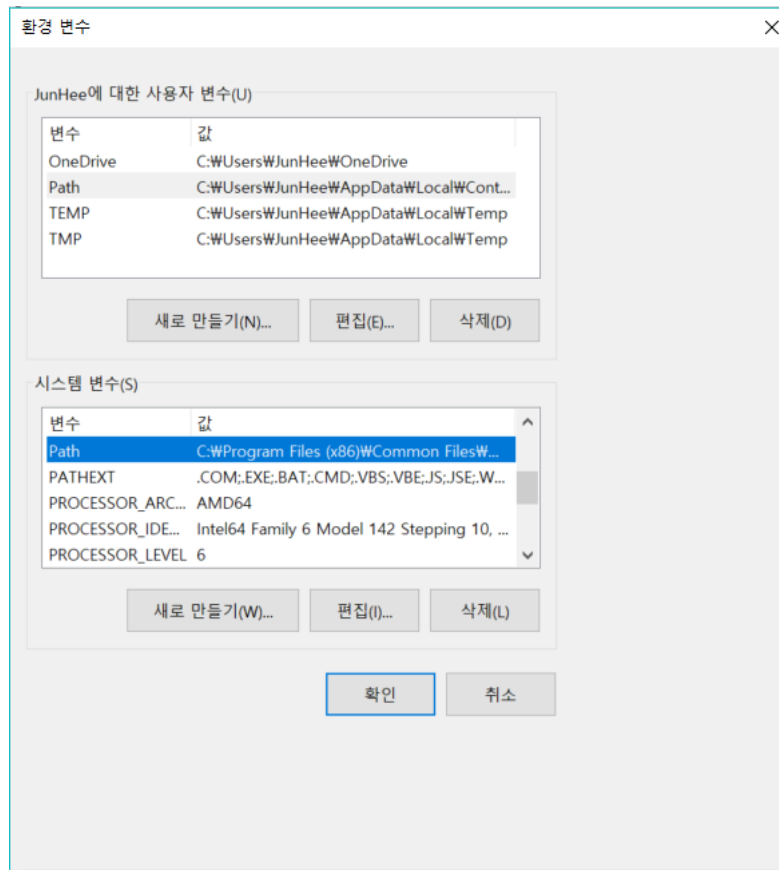


ckonlpy 설치를 위한 JAVA 설치

시스템 변수 편집 ×

변수 이름(N):	JAVA_HOME
변수 값(V):	C:\Program Files\Java\jdk1.8.0_181

ckonlpy 설치를 위한 JAVA 설치



ckonlpy 설치를 위한 JAVA 설치

시스템 변수 편집 ✕

변수 이름(N):	CLASSPATH
변수 값(V):	%JAVA_HOME%\lib

디렉터리 찾아보기(D)... 파일 찾아보기(F)... **확인** 취소

ckonlpy 설치

참고 사이트 :

https://github.com/lovit/customized_konlpy

<https://konlpy-ko.readthedocs.io/ko/v0.4.3/install/#id2>

cmd 창에서

```
pip install konlpy
```

입력 후 설치가 완료되면

```
pip install customized_konlpy
```

입력

설치에 문제가 생겼다면 연락주시면 원격으로 해결해 드리겠습니다!

ckonlpy 사용

```
from ckonlpy.tag import Twitter
```

```
tw = Twitter()
```

```
tw.add_dictionary([추가할 고유명사들], 'Noun')
```

-> ckonlpy에서 사용하는 사전에 내가 원하는 고유명사를 추가하는 함수입니다.

```
nouns_ = tw.nouns(sent_)
```

sent_라는 문장에서 noun(명사)를 뽑아내는 함수입니다.

matplotlib 설치

워드 카운팅 그래프를 그리기 위한 패키지인 matplotlib을 설치합니다.

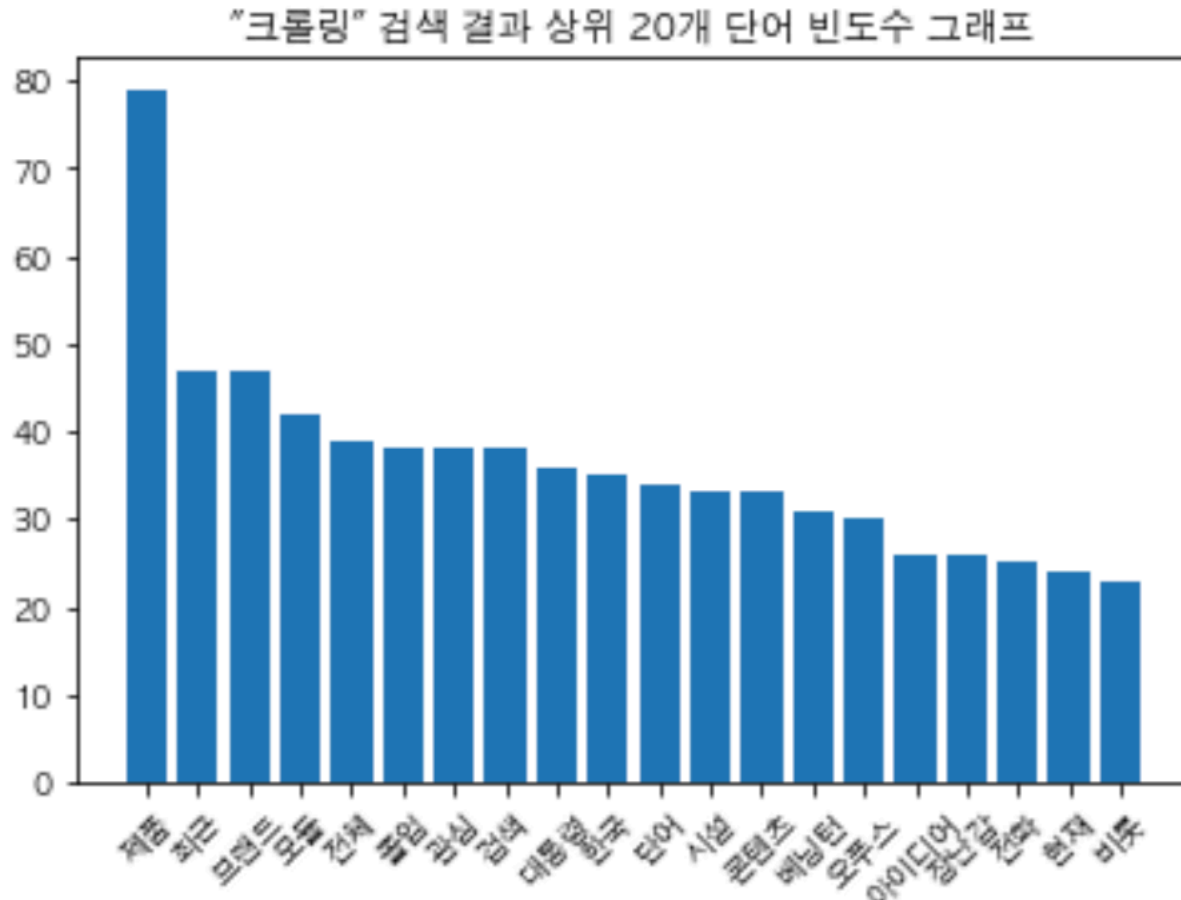
```
pip install matplotlib
```

코드에서

```
import matplotlib.pyplot as plt
```

%matplotlib -> 새 창에서 그래프가 열리게 하려면 Ipython에서 코드에 추가합니다.

matplotlib 사용



matplotlib을 사용하여 추출한 그래프 예시입니다.
다음장의 matplotlib 코드를 사용하여 그릴 수 있습니다.

matplotlib 사용

```
bar_count = 20
del_dt_sort[:20].index = range(0, len(del_dt_sort[:20]))
plt.clf()
plt.bar(range(1, bar_count+1), del_dt_sort['count'][:bar_count])
plt.xticks(range(1, bar_count+1), del_dt_sort['word'][:bar_count], rotation = 45)
plt.title('"' + keyword + '"' + ' 검색 결과 상위 ' + str(bar_count) + '개 단어 빈도수 그래프')
plt.show()
plt.savefig('word_count.png', dpi = 1000)
```

Gensim 패키지 학습의 목적.

세번째 시간에는 텍스트 데이터의 주제 분석을 위한 패키지를 학습합니다. Gensim 패키지를 사용해 텍스트데이터의 주제를 뽑아보는 실습을 진행합니다.

Gensim 패키지 설치

<https://radimrehurek.com/gensim/install.html> 참고

```
pip install --upgrade gensim
```

```
conda install -c conda-forge gensim
```

설치시 문제가 발생하면 메일 주시면 원격으로 해결해 드리겠습니다!

Gensim 패키지 사용

다음장 부터는 토픽 클러스터링을 하기 위한 코드들입니다.

코드들은 LDA를 Python으로 구현한 코드로
이론적인 배경은 아래의 URL을 참고하시면 됩니다.
(원글은 영어 텍스트 데이터를 바탕으로 짜여진 구조여서,
한글데이터 사용을 위해 조금 코드를 변경함)

[https://ratsgo.github.io/from%20frequency%20to%20semantics/
2017/07/09/lda/](https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/07/09/lda/)

Gensim 패키지 사용

```
documents_read = read_excel('nounsent.xlsx')['문장단어']

documents = []
for x_gisa in documents_read:
    while "" in x_gisa or "[" in x_gisa or "]" in x_gisa or " " in x_gisa:
        x_gisa = x_gisa.replace("", "")
        x_gisa = x_gisa.replace("[", "")
        x_gisa = x_gisa.replace("]", "")
        x_gisa = x_gisa.replace(" ", "")
    x_words = x_gisa.split(',')
    gisa = []
    for x_word in x_words:
        gisa.append(x_word)
    documents.append(gisa)
```

크롤링한 뉴스 기사를 단어단위로 다시 쪼개고, 필요없는 문장부호를 제거하는 코드입니다.

Gensim 패키지 사용

```
from collections import Counter
import random
from pandas import *

def p_topic_given_document(topic, d, alpha=0.1):
    return ((document_topic_counts[d][topic] + alpha) /
            (document_lengths[d] + K * alpha))

def p_word_given_topic(word, topic, beta=0.1):
    return ((topic_word_counts[topic][word] + beta) /
            (topic_counts[topic] + V * beta))
```


Gensim 패키지 사용

```
def topic_weight(d, word, k):  
    return p_word_given_topic(word, k) * p_topic_given_document(k, d)
```

```
def choose_new_topic(d, word):  
    return sample_from([topic_weight(d, word, k) for k in range(K)])
```

```
def sample_from(weights):  
    total = sum(weights)  
    rnd = total * random.random()  
    for i, w in enumerate(weights):  
        rnd -= w  
        if rnd <= 0:  
            return i
```

Gensim 패키지 사용

```
random.seed(0)
```

K=4

```
document_topics = [[random.randrange(K) for word in document]  
                    for document in documents]
```

```
document_topic_counts = [Counter() for _ in documents]
```

```
topic_word_counts = [Counter() for _ in range(K)]
```

```
topic_counts = [0 for _ in range(K)]
```

```
document_lengths = [len(document) for document in documents]
```

```
distinct_words = set(word for document in documents for word in document)
```

```
V = len(distinct_words)
```

```
D = len(documents)
```

Gensim 패키지 사용

```
for d in range(D):  
    for word, topic in zip(documents[d], document_topics[d]):  
        document_topic_counts[d][topic] += 1  
        topic_word_counts[topic][word] += 1  
        topic_counts[topic] += 1
```

```
for iter in range(100):  
    for d in range(D):  
        for i, (word, topic) in enumerate(zip(documents[d],  
                                                document_topics[d])):  
            document_topic_counts[d][topic] -= 1  
            topic_word_counts[topic][word] -= 1  
            topic_counts[topic] -= 1  
            document_lengths[d] -= 1  
            new_topic = choose_new_topic(d, word)  
            document_topics[d][i] = new_topic  
            document_topic_counts[d][new_topic] += 1  
            topic_word_counts[new_topic][word] += 1  
            topic_counts[new_topic] += 1  
            document_lengths[d] += 1  
print(iter)
```

Gensim 패키지 사용

```
sorted_d1 = sorted(topic_word_counts[0].items(),  
key=lambda x: x[1],reverse=True)  
sorted_d1[:10]
```

우리가 만든 첫번째 토픽의 단어들을 살펴보고 주제를 라벨링 해야하는 상황입니다. 주제는 사용자가 들어가있는 단어의 빈도와 종류를 보고 스스로 라벨링 해야합니다.

Gensim 패키지 사용

```
[ ('검색', 34), [ ('정보', 44), [ ('언급', 36), [ ('베닝턴', 31),  
  ('광고', 28), ('사용자', 42) ('대통령', 36), ('앨범', 17),  
  ('정보', 20), ('개발', 39), ('단어', 34), ('체스터', 12),  
  ('관계', 18), ('기술', 35), ('분석', 25), ('보컬', 12),  
  ('한국', 16), ('통해', 34), ('페이스북', 25), ('린킨파크', 12),  
  ('고민', 15), ('교육', 33), ('위해', 21), ('중국', 11),  
  ('중고차', 15), ('대표', 32), ('용역', 19), ('코넬', 11),  
  ('포털', 14), ('데이터', 28), ('경제', 17), ('데뷔', 11),  
  ('쇼핑', 13), ('서비스', 27), ('사업', 16), ('매뉴얼', 11),  
  ('맥주', 13)] ('창업', 26)] ('대선', 16)] ('왕홍', 10)]
```

4개로 설정한 클러스터별로 나온 단어를 보고 크롤링과 관련된 토픽을 임의로 붙여줍니다.

Ex) 1 : 검색 2 : 데이터기술 / 3 : 대선 / 4 : 차 시승 코스이름 '락 크롤링'

감사합니다.

