

[원고] 빅데이터 수집 시스템 개발

15회차 : . KoNLPy를 활용한 한국어 형태소 분석과 시각화

내용전문가	백현숙	교수설계(한기대)	홍길동
협력업체	업체명	교수설계(협력업체)	홍길동

NCS 분류 정보	20-01-02-09	정보통신 - 정보기술 - 정보기술개발 - 빅데이터플랫폼구축
능력단위 정보	03	빅데이터 수집시스템 개발
능력단위 요소 정보	2001020903_17v1.1	빅데이터 수집시스템 설계하기

업무	작성자	버전	작성일	특이사항
원고 작성	백현숙	v.1.0	2019/09/11	2회차 원고
한기대 피드백		v.1.1		
원고 보완		v.2.0		
자문 진행		v.2.1		
원고 보완				



- 학습 목차를 작성해 주세요. (NCS 비적용 과정일 경우에는 NCS 및 능력단위 정보를 기입하지 않아도 됩니다.)

NCS 분류 정보	20-01-02-09	정보통신 - 정보기술 - 정보기술개발 - 빅데이터플랫폼구축
능력단위 정보	능력단위코드 기입	빅데이터 수집시스템 개발

[illegible]

◆ 학습열기

현대사회는 정보화의 시대입니다. 정보기술이 발전함에 따라 정보의 규모는 보다 방대해지고 있으며 정보흐름의 속도 또한 폭발적으로 빨라지면서 대용량의 정보들을 신속하고 정확하게 처리하고 활용하기 위한 논의가 끊임없이 대두되고 있습니다. 이러한 정보의 처리와 활용의 대표적인 방법론으로 데이터 마이닝을 꼽을 수 있습니다. 데이터 마이닝(Data Mining)이란 의사결정 수단을 얻기 위하여 대용량의 데이터베이스(database)로부터 의미 있는 규칙과 패턴을 발견하는 기법을 말합니다(Hearst, 1999). 데이터 마이닝이 다루는 데이터는 정형화 데이터와 비정형화 데이터로 나눌 수 있습니다. 정형화 데이터란 매출데이터, 회계데이터 등과 같이 일반적으로 정형화된 수치데이터를 말합니다. 비정형화 데이터란 웹상의 포털사이트, 블로그나 SNS등 소셜미디어의 게시물과 같이 수치 데이터가 아닌 문자나 그림, 영상, 문서처럼 형태와 구조가 복잡한 데이터를 뜻합니다. 최근의 머신러닝이나 딥러닝에서는 텍스트를 분석하여 사람들의 성향이나 감정, 취향 등을 분석하여 의사결정의 수단으로 사용하거나 자동대화응답시스템이나 챗봇등에 응용하려는 움직임들이 많이 있습니다. 이 장에서는 텍스트 분석의 기본인 형태소 분석과 시각화에 대해서 다루도록 하겠습니다.

학습자가 학습 화면에 구성된 내용을 보고 있다고 가정하고, 학습 설명으로 함께 들어야 할 음성 내용을 자세하게 기입해 주세요. → 이걸 나레이션해요? 저도 이런걸 나레이션하면 좋겠어요..---

◆ 학습내용

- KoNLPy를 활용한 한국어 형태소 분석
- 시각화

◆ 학습목표

- konlpy 라이브러리를 활용하여 한국어 형태소 분석을 할 수 있다
- text 파일을 읽어서, 형태소를 나누고, 워드 클라우드를 만들 수 있다

(기입하지 않습니다.)

간지 부분

◆ KoNLPy를 활용한 한국어 형태소 분석

1. 형태소 분석 및 품사 태깅
2. konlpy 설치
3. 형태소분석 모듈 사용하기
4. 말뭉치

(기입하지 않습니다.)

◆ KoNLPy를 활용한 한국어 형태소 분석

1. 형태소 분석 및 품사 태깅

- 모든 자연언어 처리 분야에서 가장 중요하면서도 기본적으로 필요한 것이 그 언어의 형태소 분석이라 할 수 있고, 형태소 분석이 완결된 후에야 비로소 구문 분석과 의미분석을 거쳐 기계번역이라든지 자연언어 이해 시스템을 비롯한 모든 자연언어 관련 분야에 응용될 수 있다 (강승식)
- 자연어 처리에서 가장 중요한 것이 형태소 분석입니다.
- 형태소란 언어에 있어서 "최소 의미 단위"를 말합니다.
- 형태소 분석이란 형태소 보다 단위가 큰 언어 단위인 어절, 혹은 문장을 최소 의미 단위인 형태소로 분절하는 과정입니다.
- 문장을 의미 있는 작은 단위로 나누어서 주로 명사나 동사 중심으로 필요한 단어들을 추출하여 분석하는데 분석이 쉽지는 않습니다.
- 이런 분석을 도와 주는 라이브러리들이 많이 있습니다.

◆ KoNLPy를 활용한 한국어 형태소 분석

1. 형태소 분석 및 품사 태깅

형태소 분석이란 형태소를 비롯하여, 어근, 접두사/접미사, 품사(POS, part-of-speech) 등 다양한 언어적 속성의 구조를 파악하는 것입니다

품사 태깅은 형태소의 뜻과 문맥을 고려하여 그것에 마크업을 하는 일입니다

품사태깅 예시) 가방에 들어가신다 -> 가방/NNG + 에/JKM + 들어가/VV + 시/EPH + ㄴ 다/EFN

(출처: <https://konlpy-ko.readthedocs.io/ko/v0.4.3/morph/>)

1) 형태소 분석툴의 종류

종류	특징
konlpy	대표적인 한국어 형태소 분석기 java 기반이라 jdk 설치가 필요함 품사태깅, 내부에 twitter, Kkma, hannanum, komoran(현재버그문제있음) 형태소 분석기 사용 가능 기본사전으로 세종말뭉치사용 사용자 사전 등록이 가능하나 오류가 많음 , 윈도우 지원
soynlp	고유명사 추출에 유용하다 품사태깅(형태소의 뜻과 문맥을 고려하여 그것에 마크업을 하는 일)없이 토큰화 기능 우선 윈도우지원 사용자 사전 등록 안됨
mecab	은전한닢 프로젝트, 오픈소스 형태의 분석기, jdk 설치, 리눅스 기반, 윈도우 지원 어려움
khaiii(카이)	카카오에서 만든 형태소 분석기

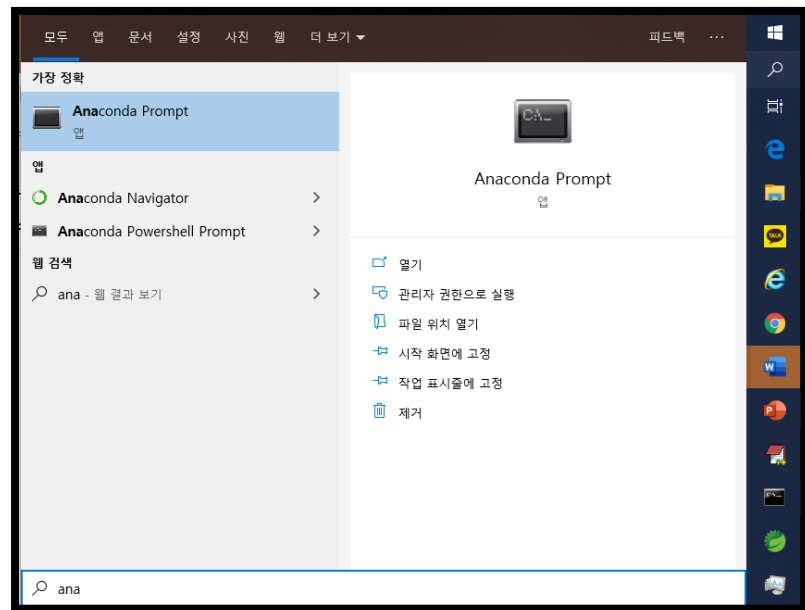
◆ OPEN JDK 다운받아 설치하기

<https://github.com/ojdkbuild/ojdkbuild>

2) konlpy 설치하기

- 먼저 JDK 를 설치하셔야 합니다.
- 시작 - 검색창에 anaconda 라고 치면
옆의 그림처럼 Anaconda Prompt 라는 메뉴가
보입니다.

이 메뉴를 선택하고 마우스 오른쪽쪽을 누른후
[관리자권한]으로 실행을 선택하세요
시스템 권한을 요하는 라이브러리도 있어서
[관리자권한]이 아닐 경우 오류가 발생할 수
있습니다.

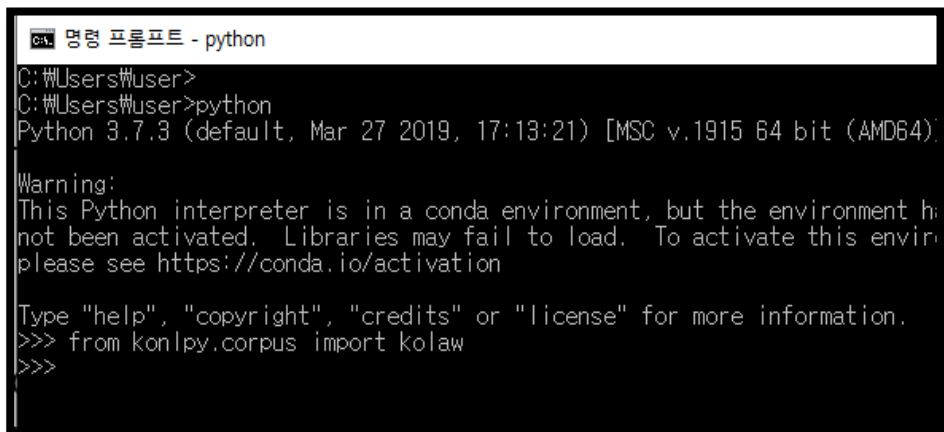


2. konlpy 설치하기

- 설치 확인

cmd창에서 python 후

from konlpy.corpus import kolaw 라고 기술하고 오류가 없다면 설치 완료입니다.



```
cmd 명령 프롬프트 - python
C:\Users\User>
C:\Users\User>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environ-
ment please see https://conda.io/activation

Type "help", "copyright", "credits" or "license()" for more information.
>>> from konlpy.corpus import kolaw
>>>
```



아나콘다 오류시 처리하기

<https://rural-mouse.tistory.com/5>

```
C:\python_workspace\파이썬데이터분석\15차시_백현숙\uni_15>python exam15_1.PY
```

```
Traceback (most recent call last):
```

```
File "exam15_1.PY", line 10, in <module>
```

```
    kkma = Kkma()
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\konlpy\tag\_kkma.py", line 95, in __init__
```

```
    jvm.init_jvm(jvmpath, max_heap_size)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\konlpy\jvm.py", line 55, in init_jvm
```

```
    jvmpath = jvmpath or jpype.getDefaultJVMPath()
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\jpype\_jvmfinder.py", line 72, in getDefaultJVMPath
```

```
    return finder.get_jvm_path()
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\jpype\_jvmfinder.py", line 209, in get_jvm_path
```

```
    .format(self._libfile))
```

```
jpype._jvmfinder.JVMNotFoundException: No JVM shared library file (jvm.dll) found. Try setting up the JAVA_HOME environment variable properly.
```

**자바가 먼저 설치되어야 하고 환경변수에 JAVA_HOME을
만들고 자바 경로를 설정해줘야 한다**
C:\Program Files\Java\jdk1.8.0_161



아나콘다 오류시

이 링크대로 아나콘다 설치 경로로 들어가서 site-packages\jpytype_jvmfinder.py에서

_get_from_java_home 모듈 부분을 보면

java_home 변수가 보인다 여기에 내가 설치한 경로를 써넣었다.

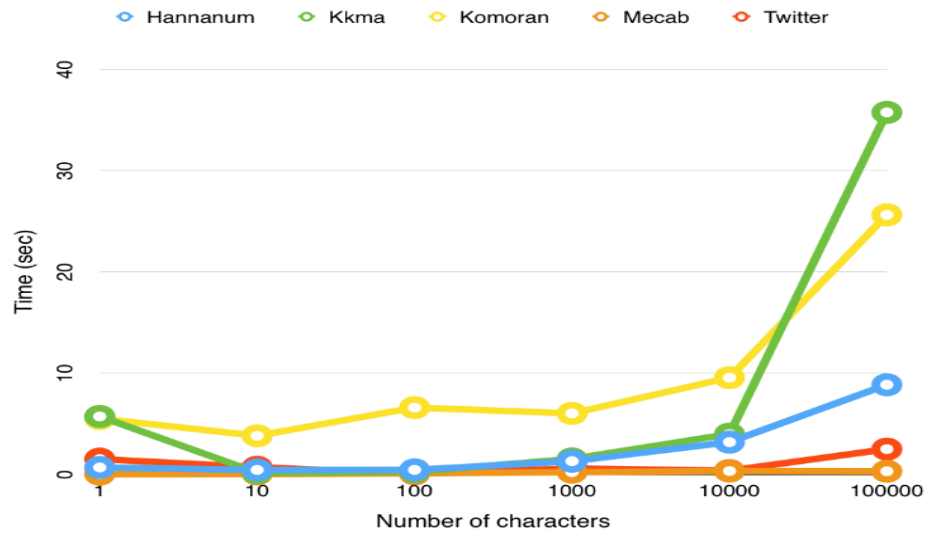
```
java_home = "C:\Program Files\Java\jdk-13.0.1"
```

3. 형태소분석 모듈 사용하기

konlpy 라이브러리에는 형태소를 분석하기 위해 지원하는 클래스가 여러개 있습니다.

- Hannanum Class(사용가능)
- Kkma Class (사용가능)
- Twitter Class (사용가능) --> Okt로 바뀜
- Komoran Class (현재 파이썬 버전에서는 작동안함)
- Mecab Class(window 7에서 지원안함)

다음은 클래스간 성능 분석 표입니다.



출처 :

1) kkma 모듈 사용하기

```
파일명 : exam15_1.py
from konlpy.tag import Kkma
from konlpy.utils import pprint
```

```
msg = """
오픈소스를 이용하여 형태소 분석을 배워봅시다. 형태소 분석을 지원하는 라이브러리가 많습니다.
각자 어떻게 분석하지는 살펴보겠습니다.
이건 Kkma모듈입니다.
"""
```

```
kkma = Kkma()
print(kkma.sentences(msg)) <- 문장으로 나눈다
print(kkma.morphs(msg) ) <- 형태소로 나눈다
print(kkma.nouns(msg)) <- 명사를 추출한다
print(kkma.pos(msg)) <- 품사태깅을 한다.
```

['오픈 소스를 이용하여 형태소 분석을 배워 봅시다.', '형태소 분석을 지원하는 라이브러리가 많습니다.', '각자 어떻게 분석하지는 살펴보겠습니다.', '이건 Kkma 모듈입니다.']

['오픈', '소스', '를', '이용', '하', '여', '형태소', '분석', '을', '배우', '어', '보', '보시다', '.', '형태소', '분석', '을', '지원', '하', '는', '라이브러리', '가', '많', '습니다', '.', ':', '각', '자', '어떻', '게', '분석', '하', '지', '는', '살펴보', '겠', '습니다', '.', ':', '이건', 'Kkma', '모듈', '이', '보니다', '.']

['오픈', '오픈소스', '소스', '이용', '형태소', '분석', '지원', '라이브러리', '이건', '모듈']

[('오픈', 'NNG'), ('소스', 'NNG'), ('를', 'JKO'), ('이용', 'NNG'), ('하', 'XSV'), ('여', 'ECS'), ('형태소', 'NNG'), ('분석', 'NNG'), ('을', 'JKO'), ('배우', 'V'), ('어', 'ECS'), ('보', 'VV'), ('보시다', 'EFA'), ('.', 'SF'), ('형태소', 'NNG'), ('분석', 'NNG'), ('을', 'JKO'), ('지원', 'NNG'), ('하', 'XSV'), ('는', 'ETD'), ('라이브러리', 'NNG'), ('가', 'JKS'), ('많', 'VA'), ('습니다', 'EFN'), ('.', 'SF'), ('각', 'V'), ('자', 'ECE'), ('어떻', 'VA'), ('게', 'ECD'), ('분석', 'NNG'), ('하', 'XSV'), ('지', 'ECD'), ('는', 'JK'), ('살펴보', 'V'), ('겠', 'EPT'), ('습니다', 'EFN'), ('.', 'SF'), ('이건', 'NIP'), ('Kkma', 'OL'), ('모듈', 'NNG'), ('이', 'VCP'), ('보니다', 'EFN'), ('.', 'SF')]

2)Hannanum 모듈 사용하기

파일명 : exam15_2.py

```

from konlpy.tag import Hannanum
from konlpy.utils import pprint

```

```

msg = """
오픈소스를 이용하여 형태소 분석을 배워봅시다.
각자 어떻게 분석하지는 살펴보겠습니다.
이건 Hannanum 모듈입니다.
"""

hannanum = Hannanum()
print(hannanum.analyze(msg))
print(hannanum.morphs(msg))
print(hannanum.nouns(msg))
print(hannanum.pos(msg))

[('오픈', 'ncpa'), ('소', 'ncn'), ('들', 'jco')], [('이용', 'ncpa'), ('하', 'xsva'), ('어', 'ecs')], [('이용', 'ncpa'), ('하', 'xsva'), ('어', 'ecx')], [('이용', 'ncpa'), ('하', 'xsva'), ('어', 'ef')], [('형태소', 'ncn')], [('형태', 'ncn'), ('소', 'ncn')], [('분석', 'ncpa'), ('을', 'jco')], [('배우', 'pvg'), ('어', 'ecx'), ('보', 'px'), ('보다', 'ef')], [('.', 'sf')], [('.', 'sy')], [], [('형태소', 'ncn')], [('형태', 'ncn'), ('소', 'ncn')], [('분석', 'ncpa'), ('을', 'jco')], [('지원', 'ncpa'), ('하', 'xsva'), ('는', 'etm')], [('지원', 'ncpa'), ('하', 'xsva'), ('어', 'ecs'), ('는', 'jxc')], [('지원', 'ncpa'), ('하', 'xsva'), ('어', 'ef'), ('는', 'etm')], [('라이브러리', 'ncn'), ('가', 'jcc')], [('라이브러리', 'ncn'), ('가', 'jcs')], [('말', 'paa'), ('습니다', 'ef')], [('.', 'sf')], [('.', 'sy')], [], [('각자', 'mag')], [('각자', 'ncn')], [('어떻게', 'pad'), ('게', 'ecs')], [('어떻게', 'pad'), ('게', 'ecx')], [('어떻게', 'mag')], [('분석', 'ncpa'), ('하', 'xsva'), ('지는', 'ecs')], [('분석', 'ncpa'), ('하', 'xsva'), ('지는', 'ecx')], [('분석', 'ncpa'), ('하', 'xsva'), ('지는', 'etm')], [('분석', 'ncpa'), ('하', 'xsva'), ('지', 'ecs'), ('는', 'jxc')], [('분석', 'ncpa'), ('하', 'xsva'), ('지', 'ef'), ('는', 'etm')], [('분석', 'ncpa'), ('하', 'xsva'), ('어', 'ecx'), ('지', 'px'), ('는', 'etm')], [('분석', 'ncpa'), ('하', 'xsva'), ('어', 'ef'), ('지', 'etm')], [('살펴보', 'pvg'), ('겠습니다', 'ef')], [('살피', 'pvg'), ('어', 'ecx'), ('보', 'px'), ('겠', 'ep'), ('습니다', 'ef')], [('.', 'sf')], [('.', 'sy')], [], [('이', 'ncn'), ('건', 'xsnx')], [('이', 'ncn'), ('이', 'jp'), ('건', 'ecc')], [('이', 'ncn'), ('이', 'jp'), ('건', 'ecs')], [('이', 'nnc'), ('건', 'xsnx'), ('이', 'nnc'), ('건', 'xsnx')], [('이', 'nnc'), ('건', 'xsnx')], [('이', 'nnc'), ('이', 'jp'), ('건', 'ecc')], [('이', 'nnc'), ('이', 'jp'), ('건', 'ecs')], [('이', 'pvg'), ('건', 'ecc')], [('이', 'pvg'), ('건', 'ecs')], [('Hannanum', 'f')], [('모듈', 'ncn'), ('이', 'jp'), ('보다', 'ef')], [('모듈입니다', 'ncn'), ('이', 'jp'), ('다', 'ef')], [('모듈입니다', 'nqq'), ('이', 'jp'), ('다', 'ef')], [('.', 'sf')], [('.', 'sy')]]

[('오픈소스', '들', '이용', '하', '어', '형태소', '분석', '을', '배우', '어', '보', '보다', '.', '형태소', '분석', '을', '지원', '하', '는', '라이브러리', '가', '말', '습니다', '.', '각자', '어떻게', '게', '분석', '하', '어', '지', '는', '살피', '어', '보', '겠습니다', '.', '이', '이', '건', 'Hannanum', '모듈입니다', '이', '다', '.'), ('오픈소스', '이용', '형태소', '분석', '형태소', '분석', '지원', '라이브러리', '각자', '분석', '이', '모듈입니다'), ('오픈소스', 'N'), ('들', 'J'), ('이용', 'N'), ('하', 'X'), ('어', 'E'), ('형태소', 'N'), ('분석', 'N'), ('을', 'J'), ('배우', 'P'), ('어', 'E'), ('보', 'P'), ('보다', 'E'), ('.', 'S'), ('형태소', 'N'), ('분석', 'N'), ('을', 'J'), ('지', 'N'), ('는', 'E'), ('라이브러리', 'N'), ('가', 'J'), ('말', 'P'), ('습니다', 'E'), ('.', 'S'), ('각자', 'N'), ('어떻게', 'P'), ('게', 'E'), ('분석', 'N'), ('하', 'X'), ('어', 'E'), ('지', 'P'), ('는', 'E'), ('살피', 'P'), ('어', 'E'), ('보', 'P'), ('겠습니다', 'E'), ('.', 'S'), ('이', 'N'), ('이', 'J'), ('건', 'E'), ('Hannanum', 'F'), ('모듈입니다', 'N'), ('이', 'J'), ('다', 'E'), ('.', 'S')]]

```

3)Twitter (Okt로 변경됨) 모듈 사용하기

파일명 : exam15_3.py

```
from konlpy.tag import Kkma, Hannanum, Komoran, Mecab, Okt
from konlpy.utils import pprint
```

```
msg = """
오픈소스를 이용하여 형태소 분석을 배워봅시다. 형태소 분석을 지원하는 라이브러리가 많습니다.
각자 어떻게 분석하지는 살펴보겠습니다.
이건 Twitter 모듈입니다.
"""
```

```
twitter = Okt()
print(twitter.morphs(msg))
print(twitter.nouns(msg))
print(twitter.pos(msg))
```

```
['\n', '오픈소스', '를', '이용', '하여', '형태소', '분석', '을', '배워', '봅시다', '.', '형태소', '분석', '을', '지원', '하는', '라이브러리', '가', '많습니다', '.',
'각자', '어떻게', '분석', '하', '지는', '살펴보겠습니다', '.', '이건', 'Twitter', '모듈', '입니다', '.', '\n']
['오픈소스', '이용', '형태소', '분석', '형태소', '분석', '지원', '라이브러리', '각자', '분석', '이건', '모듈']
[(('\n', 'Foreign'), ('오픈소스', 'Noun'), ('를', 'Josa'), ('이용', 'Noun'), ('하여', 'Verb'), ('형태소', 'Noun'), ('분석', 'Noun'), ('을', 'Josa'), ('배워', 'Verb'),
('봅시다', 'Verb'), ('.', 'Punctuation'), ('형태소', 'Noun'), ('분석', 'Noun'), ('을', 'Josa'), ('지원', 'Noun'), ('하는', 'Verb'), ('라이브러리', 'Noun'), ('가', '
Josa'), ('많습니다', 'Adjective'), ('.', 'Punctuation'), ('각자', 'Noun'), ('어떻게', 'Adjective'), ('분석', 'Noun'), ('하', 'Suffix'), ('지는', 'Josa'), ('살펴보겠
습니다', 'Verb'), ('.', 'Punctuation'), ('이건', 'Noun'), ('Twitter', 'Alpha'), ('모듈', 'Noun'), ('입니다', 'Adjective'), ('.', 'Punctuation'), ('\n', 'Foreign'))]
```

4) 파일을 읽어서 명사로 분리하기 예제

파일명 : exam15_4.py

```
from konlpy.tag import Kkma, Hannanum, Komoran, Mecab, Okt
from konlpy.utils import pprint

file = open("./data/data1.txt")
msg = file.read()

print("---- Kkma 클래스 -----")
kkma = Kkma()
print("---- 문장으로 분해 ---- ")
print(kkma.sentences(msg))

print("---- 명사분해 ---- ")
print(kkma.nouns(msg))

print("---- 품사태깅 ---- ")
print(kkma.pos(msg))
```

```
hannanum = Hannanum()
print("---- 문장으로 분해 ---- ")
print(hannanum.analyze(msg))
print("---- 형태소로 분해 ---- ")
print(hannanum.morphs(msg))
print("---- 명사분해 ---- ")
print(hannanum.nouns(msg))
print("---- 품사태깅 ---- ")
print(hannanum.pos(msg))

twitter = Twitter()
print("---- Mecab 클래스 -----")
twitter = Twitter()
print("---- 형태소로 분해 ---- ")
print(twitter.morphs(msg))
print("---- 명사분해 ---- ")
print(twitter.nouns(msg))
print("---- 품사태깅 ---- ")
print(twitter.pos(msg))
```

4. 말뭉치와 사전

- 말뭉치란, 언어 연구를 위해 텍스트를 컴퓨터가 읽을 수 있는 형태로 모아 놓은 언어 자료이다.
- 통계 분석 및 [가설 검증](#)을 수행하거나, 특정한 언어 영역 내에서 언어 규칙 발생의 검사와 그 규칙의 정당성 입증에 사용된다
- 어떤 언어를 분석함에 있어서 기준이 되는 집합인데 이 집합을 어떤 걸 사용 하느냐에 따라 분석 결과가 달라진다
- 만일 법과 관련된 언어들을 분석하려면 법률관련 말뭉치가 필요하고 예술과 관련하여 언어들을 분석하려면 예술관련 말뭉치가 있어야 한다
- konlpy 라이브러리에서 제공되는 말뭉치 들에는 다음과 같은 것들이 있습니다.
 - kolaw : 한국 법률 말뭉치, constitution.txt
 - kobill : 대한민국 국회 의안 말뭉치. 파일 ID는 의안 번호를 의미합니다. 1809890.txt - 1809899.txt
 - 세종말뭉치 : 국립국악원에서 모아놓은 국어 말뭉치
 - kaist 말뭉치 : kiast에서 만들었음
- 사전은 말뭉치를 이용해 구축되었고, 이 사전들이 형태소분석이나 품사 태깅에 사용됩니다
 - Hannanum 사전 : kist 말뭉치를 이용해 구축되었다
 - Kkma 사전 : 세종 말뭉치를 이용해 구축되었다
 - Mecab 사전 : 세종말뭉치를 이용해 구축

과정명	모듈명	회차명
-----	-----	-----

4. 말뭉치와 사전

```
파일명 : exam15_5.py
from konlpy.corpus import kolaw, kobill
wordList = kolaw.open('constitution.txt').readlines()
print(wordList[:3]) #3줄만 출력
print()
wordList2 = kobill.open('1809890.txt').readlines()
print(wordList2[:30]) #3줄만 출력
```

['대한민국헌법\n', '\n', '유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며, 자율과 조화를 바탕으로 자유민주적 기본질서를 더욱 확고히 하여 정치·경제·사회·문화의 모든 영역에 있어서 각인의 기회를 균등히 하고, 능력을 최고도로 발휘하게 하며, 자유와 권리에 따르는 책임과 의무를 완수하게 하여, 안으로는 국민생활의 균등한 향상을 기하고 밖으로는 항구적인 세계평화와 인류공영에 이바지함으로써 우리들과 우리들의 자손의 안전과 자유와 행복을 영원히 확보할 것을 다짐하면서 1948년 7월 12일에 제정되고 8차에 걸쳐 개정된 헌법을 이제 국회의 의결을 거쳐 국민투표에 의하여 개정한다.\n']

['지방공무원법 일부개정법률안\n', '\n', '(정의화의원 대표발의)\n', '\n', ' 의 안\n', ' 번 호\n', '\n', '9890\n', '\n', '발의연월일 : 2010. 11. 12. \n', '\n', ' 발 의 자 : 정의화,이명수,김을동 \n', '\n', '이사철,여상규,안규백\n', '\n', '황영철,박영아,김정훈\n', '\n', '김학송 의원(10인)\n', '\n', '제안이유 및 주요내용\n', '\n', ' 초등학교 저학년의 경우에도 부모의 따뜻한 사랑과 보살핌이 필요\n', '\n', '한 나이이나, 현재 공무원이 자녀를 양육하기 위하여 육아휴직을 할 \n', '\n', '수 있는 자녀의 나이는 만 6세 이하로 되어 있어 초등학교 저학년인 \n', '\n', '자녀를 돌보기 위해서는 해당 부모님은 일자리를 그만 두어야 하고 \n', '\n', '이는 곧 출산의욕을 저하시키는 문제로 이어질 수 있을 것임.\n']

내레이션을	
-------	--

22

과정명	모듈명	회차명
-----	-----	-----

간지 부분

◆ 시각화

1. 필요한 모듈 설치하기
2. wordcloud 작성하기
3. 한글폰트
4. 파일에서 읽은 데이터를 시각화하기
5. 워드클라우드 다른 라이브러리 사용하기
6. 다양한 모양의 워드클라우드 만들기

(기입하지 않습니다.)

과정명	모듈명	회차명
-----	-----	-----

1. 필요한 모듈 설치하기

```

pip install pytagcloud <- 워드클라우드 지원 라이브러리
pip install pygame

pip install WordCloud <- 워드 클라우드 지원 라이브러리(만든사람이 다름)

```

2. wordcloud 작성하기

pytagcloud 라이브러리는 데이터를 단어와 단어의 빈도수를 tuple 로 묶어서 list 형태로 전달해야 합니다.

예) [('school',30), ('rainbow',10), ('cloud',23), ('peach',10), ('pink',20)]

파일명 : exam15_6.py
import pytagcloud #wordcloud 라이브러리
import webbrowser #브라우저 제어 라이브러리

```
tag = [( ' school ',30), ( ' rainbow ',10), ( ' cloud ',23), ( ' peach ',10), ( ' pink ',20)] #데이터 준비
#print(tag)
taglist = pytagcloud.make_tags(tag, maxsize=50)
#각 단어를 {'color': (131, 194, 49), 'size': 68, 'tag': 'school'} 색, 크기, tag형태의 dict타입으로 만든다,

#print(taglist)
pytagcloud.create_tag_image(taglist,
                             'wordcloud.jpg', #wordcloud.jpg 이름으로 저장한다
                             size=(300, 300), #이미지 크기
                             fontname=' Nobile', #폰트명
                             rectangular=True) #이미지 모양, true 일 경우 사각형으로 그린다

#브라우저에 이미지를 보여준다
webbrowser.open('wordcloud.jpg')
```

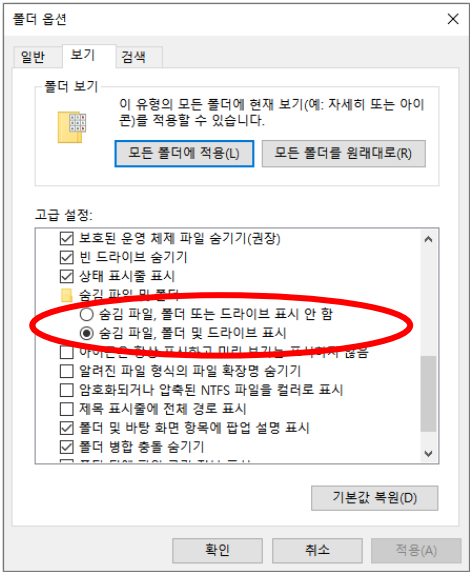


3. 한글폰트

- pytagcloud 는 한글 폰트를 지원하지 않습니다. 그래서 한글을 쓰려면 아래와 같이 폰트를 별도로 설치하여야 합니다.
- C:\ProgramData\Anaconda3\Lib\site-packages\pytagcloud\fonts 이 경로에 한글 폰트가 있어야 합니다.
- 한글 폰트는 웹사이트에 원하는 폰트를 다운받아도 되고 C:/Windows/Fonts 에 보면 많이 있습니다.
- 위 폴더에서 한글을 지원하는 폰트 파일들을 복사하여 C:\ProgramData\Anaconda3\Lib\site-packages\pytagcloud\fonts 경로에 붙여넣기를 합니다.
- ProgramData폴더는 중요한 시스템 폴더라 c:/에서 이 폴더가 보이지 않습니다. 위 경로를 직접 모두 치면 됩니다

3. 한글폰트

- 만일 폴더를 보기를 원하면, 탐색기의 보기메뉴에서 - 옵션 - 폴더및 검색옵션을 선택 후 보기 탭을 누른 후 숨김파일, 폴더및 드라이브 표시를 선택해주면 됩니다.



과정명	모듈명	회차명
-----	-----	-----

3. 한글폰트

C:\ProgramData\Anaconda3\Lib\site-packages\pytagcloud\fonts 에 있는 fonts.json 파일을 visual studio code 에서 열고 맨 앞에 한글 폰트를 추가해야 합니다.

예시)

```
{
    "name": "HangleFont1",
    "ttf": "H2GTRE.TTF",
    "web": "http://fonts.googleapis.com/css?family=Nobile"
},
```

이름은 제 마음대로 정하면 됩니다. ttf에는 복사해온 한글폰트 파일명을 작성하면 됩니다. 여러분 컴퓨터에 저 폰트가 없을 수도 있습니다. 가지고 있는 한글 폰트를 사용하면 됩니다. web은 현재 위치에 폰트가 없을 경우 web으로부터 다운받으라는 url인데 그냥 아래에 있는 web url 하나를 복사해 왔습니다. 파일의 내용을 삭제하는게 아니라 위에 추가하는 겁니다

내 레이 션	
--------------	--

과정명	모듈명	회차명
-----	-----	-----

3. 한글폰트

C:\ProgramData\Anaconda3\Lib\site-packages\pytagcloud\fonts 에 있는 fonts.json 파일을 visual studio code 에서 열고 맨 앞에 한글 폰트를 추가해야 합니다.

예시)

```
{
    "name": "HangleFont1",
    "ttf": "H2GTRE.TTF",
    "web": "http://fonts.googleapis.com/css?family=Nobile"
},
```

이름은 제 마음대로 정하면 됩니다. ttf에는 복사해온 한글폰트 파일명을 작성하면 됩니다. 여러분 컴퓨터에 저 폰트가 없을 수도 있습니다. 가지고 있는 한글 폰트를 사용하면 됩니다. web은 현재 위치에 폰트가 없을 경우 web으로부터 다운받으라는 url인데 그냥 아래에 있는 web url 하나를 복사해 왔습니다. 파일의 내용을 삭제하는게 아니라 위에 추가하는 겁니다

내 레이 션	
--------------	--

3. 한글폰트

파일명 : exam15_7.py

```
import pytagcloud
import webbrowser

tag = [('학교',30), ('무지개',10), ('구름',23), ('복숭아',14), ('분홍색',20)]
#print(tag)
taglist = pytagcloud.make_tags(tag, maxsize=40)

#print(taglist)
pytagcloud.create_tag_image(taglist,
    'wordcloud.jpg',
    size=(300, 300),
    fontname='HangleFont1', rectangular=True)

#브라우저에 이미지를 보여준다
webbrowser.open('wordcloud.jpg')
```

이미지나 색은 계속 랜덤하게 바뀝니다.



5. 워드클라우드 다른 라이브러리 사용하기

- wordcloud 라는 라이브러리가 있는데 앞에서 보았던 pytagcloud 모듈 보다 사용하기가 편합니다.
- 별도로 Counter 클래스를 사용하지 않고 text를 전달하면 형태소를 분석하여 시각화를 진행합니다 .
- 우선 라이브러리를 설치합니다.

pip install wordcloud

```
import matplotlib.pyplot as plt  
from wordcloud import WordCloud
```

```
wordcloud = WordCloud().generate(text)
```

```
plt.imshow(wordcloud, interpolation='bilinear')  
plt.axis("off")  
plt.show()
```

5. 워드클라우드 다른 라이브러리 사용하기

파일명 : exam15_9.py

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

#1.txt 파일 읽기

```
file = open("./data/alice.txt")
```

```
text = file.read()
```

#2. text 를 그대로 전달한다

```
wordcloud = WordCloud().generate(text)
```

```
plt.imshow(wordcloud, interpolation='bilinear')
```

```
plt.axis("off")
```

```
plt.show()
```



```

파일명 : exam15_10.py
from os import path
from wordcloud import WordCloud
import nltk
from matplotlib import font_manager, rc, matplotlib_fname
import matplotlib.pyplot as plt

#한국 법률 말뭉치
from konlpy.corpus import kolaw
from konlpy.tag import Twitter

#한국법률말뭉치로 부터 헌법을 읽어온다
c = kolaw.open('constitution.txt').read()
print(c)

# Generate a word cloud image
wordcloud = WordCloud(font_path="c:/Windows/Fonts/malgun.ttf").generate(c)
wordcloud.to_file("image4.png")

# Display the generated image:
# the matplotlib way:

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

```



```

파일명 : exam15_10.py
from os import path
from wordcloud import WordCloud
import nltk
from matplotlib import font_manager, rc, matplotlib_fname
import matplotlib.pyplot as plt

#한국 법률 말뭉치
from konlpy.corpus import kolaw
from konlpy.tag import Twitter

#한국법률말뭉치로 부터 헌법을 읽어온다
c = kolaw.open('constitution.txt').read()
print(c)

wordcloud = WordCloud(font_path="c:/Windows/Fonts/malgun.ttf").generate(c)
wordcloud.to_file("image4.png")

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

```



```

파일명 : exam15_10.py
from os import path
from wordcloud import WordCloud
import nltk
from matplotlib import font_manager, rc, matplotlib_fname
import matplotlib.pyplot as plt

#한국 법률 말뭉치
from konlpy.corpus import kolaw
from konlpy.tag import Twitter

#한국법률말뭉치로 부터 헌법을 읽어온다
c = kolaw.open('constitution.txt').read()
print(c)

# Generate a word cloud image
wordcloud = WordCloud(font_path="c:/Windows/Fonts/malgun.ttf").generate(c)
wordcloud.to_file("image4.png")

# Display the generated image:
# the matplotlib way:

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

```



6. 다양한 모양의 워드클라우드 만들기

- wordcloud 라이브러리는 이미지에 맞추어서 다양한 형태의 워드클라우드를 만들 수 있습니다.

1. 이미지를 준비한다 , 이미지를 numpy 배열로 변환한다

```
mask_image = np.array(Image.open("./1.jpg"))
```

2. WordCloud 의 mask속성에 이 배열을 전달한다

```
wordcloud = WordCloud(font_path="c:/Windows/Fonts/malgun.ttf",  
    relative_scaling=0.2,  
    background_color='white',  
    mask=mask_image,).generate_from_frequencies(temp_data)
```

(기입하지 않습니다.)

6. 다양한 모양의 워드클라우드 만들기

```
from wordcloud import WordCloud
import nltk
from collections import Counter
#한국 법률 말뭉치
from konlpy.corpus import kolaw
from konlpy.tag import Twitter
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import ImageColorGenerator
from PIL import Image
import random
```

```
c = open("./data/data1.txt", 'r').read()
#명사만 골라낸다
t = Twitter()
tokens_ko = t.nouns(c)
print(tokens_ko)
```

```
#제거하고 싶은 단어들 제거하기
stop_words = ['다만', '그', '곳', '나', '일', '패', '달리', '의']
tokens_ko = [each_word for each_word in tokens_ko if each_word not in stop_words]
#print(tokens_ko)
```



```
data = Counter(tokens_ko).most_common(100)
temp_data = dict(data)

mask_image = np.array(Image.open("./image/1.jpg")) #image 폴더 아래의 1.jpg 이미지를 불러온다

wordcloud = WordCloud(font_path= " c:/Windows/Fonts/malgun.ttf " ,
    relative_scaling=0.2,
    background_color= ' white ' ,
    mask=mask_image).generate_from_frequencies(temp_data)

plt.figure(figsize=(16,8)) #차트 화면의 크기를 지정한다
random.seed(1234) #색 정보가 랜덤하게 생성되기 때문에 계속 바뀌는걸 방지하기 위해 값을 주었다. 1234는 임의로 준 값임
#별도의 색을 부여하려면 사용자 정의 함수를 작성하여 전달해야 한다
def custom_color_func(word, font_size, position, orientation, random_state=None,
    **kwargs):

    color= " rgb({},{},{}) ".format( random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    # rgb 함수는 red, green, blue 의 세가지 색을 섞어서 색을 만든다. 값은 0~255까지만 부여가 가능하다
    return color

#사용자 정의 함수를 전달했다
plt.imshow(wordcloud.recolor(color_func=custom_color_func), interpolation= " bilinear " )

plt.axis("off")
plt.show()
```


◆ 적용하기- 풀이

```

from wordcloud import WordCloud
import nltk
from collections import Counter
#한국 법률 말뭉치
from konlpy.corpus import kolaw
from konlpy.tag import Twitter
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import ImageColorGenerator
from PIL import Image
import random

#한국법률말뭉치로 부터 헌법을 읽어온다
c = kolaw.open('constitution.txt').read()
print(c)

t = Twitter()
tokens_ko = t.nouns(c)
print(tokens_ko)

#제거하고 싶은 단어들 제거하기
stop_words = ['다만', '그', '곳', '나', '일', '패', '달리', '의']
tokens_ko = [each_word for each_word in tokens_ko if each_word not in stop_words]
#print(tokens_ko)

```

(기입하지 않습니다.)

◆ 적용하기 -> 풀이

```
data = Counter(tokens_ko).most_common(100)
temp_data = dict(data)

mask_image = np.array(Image.open("./image/kroea_mask.jpg"))

wordcloud = WordCloud(font_path="c:/Windows/Fonts/malgun.ttf",
                      relative_scaling=0.2,
                      background_color='white',
                      mask=mask_image,).generate_from_frequencies(temp_data)

plt.figure(figsize=(16,8))

random.seed(1234)
#별도의 색을 부여하려면
def custom_color_func(word, font_size, position, orientation, random_state=None,
                      **kwargs):

    color="rgb({}, {}, {})".format( random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    return color

plt.imshow(wordcloud.recolor(color_func=custom_color_func), interpolation="bilinear")

plt.axis("off")
plt.show()
```

(기입하지 않습니다.)

◆ 문제풀기

문제 1) 형태소 분석에 대하여 잘못된 설명은 ?

- ① 형태소란 언어에 있어서 "최소 의미 단위"를 말합니다
- ② 자양어 처리에서 가장 중요합니다.
- ③ 말뭉치들로부터 사전을 구성하여 형태소 분석에 이용합니다.
- ④ 형태소 분석에 어떤 사전을 사용하는 지는 중요하지 않습니다.

정답) 4

해설) 분석할 대상에 다 맞는 사전은 없습니다. 법률관련 텍스트 분석에는 법률사전, 영화관련 텍스트 분석에는 영화관련사전등 관련된 사전을 사용해야 합니다.

난이도) 5(1:아주어려움, 2:어려움 3: 보통 4: 쉬움 5: 아주 쉬움)

관련페이지번호) 6

(기입하지 않습니다.)

번 이	문제	정답	난이도	해설	관련학습보기
2	<p>다음 중 윈도우를 지원하지 않는 형태소 분석 클래스는?</p> <p>① KKma</p> <p>② Twitter</p> <p>③ Hannanum</p> <p>④ Mecab</p>	4	4	mecab 은 윈도우를 지원하지 않습니다.	11
3	<p>KKma 모듈이 지원하는 형태소 분석 함수의 기능으로 잘못 연결된것은?</p> <p>① kkma.sentences(msg) - 문장으로 나눈다</p> <p>② kkma.morphs(msg) - 형태로 나눈다</p> <p>① kkma.nouns(msg) - 명사를 추출한다</p> <p>② kkma.pos(msg) - 명사를 추출한다</p>	4	5	4번은 품사태깅을 합니다	13
4	<p>다음 단어들을 pytagcloud 라이브러리 전달하고자 할때 기술방식은?</p> <p>클라우드, 10</p> <p>워드, 20</p> <p>형태소, 30</p>	해답참조	4	[('클라우드',10), (' 워드 ',20), (' 형태소 ',30)]	

번호	문제	정답	난이도	해설	관련학습 보기
5	random.seed 함수의 역할은?	해설참조	2	차트의 색이 계속 랜덤하게 바뀌는걸 막기 위해서 사용합니다.	37

◆ 핵심요약

- ▶ KoNLPy를 활용한 한국어 형태소 분석
 - 형태소 분석이란 형태소 보다 단위가 큰 언어 단위인 어절, 혹은 문장을 최소 의미 단위인 형태소로 분절하는 과정입니다
 - konlpy 라이브러리는 한글 형태소를 분석하기 위해 카이스트에서 만든 라이브러리입니다
 - konlpy는 Kkma, Twitter, Hannanum, Komoran 등의 tag 패키지를 지원합니다.
- ▶ 시각화
 - 워드클라우드는 단어를 분석해 시각화 하는 차트로 가장 인기 있는 차트입니다.
 - pytagcloud, wordcloud 등의 모듈이 있습니다.
 - 두 모듈 모두 문장을 형태소 분석하여 단어와 빈도수의 데이터 타입을 만들어 전달해야 합니다.
 - wordcloud 라이브러리는 이미지 마스킹 기능을 이용해 특정한 이미지 모양에 맞는 윈드 클라우드를 만들 수 있고 사용자가 원하는 색상을 부여할 수 있습니다.
 -

(기입하지 않습니다.)

과정명	모듈명	회차명
-----	-----	-----

◆ 학습맺음

- 이번 회차에서는 DataFrame 의 구조와 DataFrame이 제공하는 API를 이용해서 외부파일을 읽고 쓰는 방법, 데이터 분석 API 등을 살펴보았습니다
수고 많으셨습니다.

◆ 참고자료

- <https://docs.anaconda.com/anaconda/>
- <https://code.visualstudio.com/docs/getstarted/themes>
- <https://konlpy-ko.readthedocs.io/ko/v0.4.3/>
- <https://m.blog.naver.com/PostView.nhn?blogId=2035icck&logNo=220784000304&proxyReferer=https%3A%2F%2Fwww.google.com%2F>
- <https://ko.wikipedia.org/wiki/%EB%A7%90%EB%AD%89%EC%B9%98>
- https://amueller.github.io/word_cloud/index.html

(기입하지 않습니다.)

회차 메타데이터

- 주요학습내용 : 해당 회차 또는 레슨의 주요학습내용을 자세히 기입해 주세요.
- 검색 키워드 : 학습자가 검색창에 어떤 검색어를 입력하면 본 회차 또는 본 레슨이 검색될 수 있을지 검색 키워드를 5개 기입해 주세요.

제목	주요학습내용	검색 키워드1	검색 키워드2	검색 키워드3	검색 키워드4	검색 키워드5
15. KoNLPy를 활용한 한국어 형태소 분석과 시각화	KoNLPy 라이브러리를 이용하여 텍스트 파일을 분석하여 시각화하기	형태소분석	Konlpy	wordcloud	Counter	Twitter