자연어 처리

설치 목록 (설치 순서 반드시 지킬 것!!!)

JDK (Java SE Downloads)
JAVA_HOME 설정
JPype 설치
KoNLPy 설치
Word Cloud 설치

JDK 설치 : Java JDK로 검색해서 OS에 맞춰 설치
JAVA_HOME 설정 :

JPype1 : conda install -c conda-forge jpype1
gensim install : pip install gensim
KoNLPy : pip install konlpy

WordCloud 설치 : pip install wordcloud

- ### JPype

Python 으로 하여금 거의 모든 Java 라이브러리를 사용하게 한다.
pip install JPype1-py3
conda install -c conda-forge jpype1

- ### gensim

text 데이터들의 topic modelling 하는 라이브러리.
즉 topic modelling 에 대한 여러가지 기능이 구현되어 있다.
topic model (topic은 주제라는 뜻.)
예를 들면
"트와이스는 너무 좋고 나는 모모를 좋아 한다. " 라는 문장이 있으면
이 문장의 주제는 트와이스 혹은 모모 일것이다.
그것은 관점에 따라 좀 다른데
일반적으로 사람은 이 문장의 주제가 무엇일까? 라고 질문하면

트와이스 라고 이야기 할 관점이 크고
그다음에 “트와이스 멤버의 모모” 라고 이야기 하는사람도 있을 것이다.
이런식으로 텍스트 데이터 집단에서 해당 텍스트 집단들의 주제를 추출할수 있거나
만들수 있는 모델을 topic modelling 이라고 할수 있다.

한글 자연어처리 기초

Kkma 꼬꼬마

```
from lib2to3.btm_utils import tokens

from konlpy.tag import Kkma

kkma = Kkma()

print(kkma.sentences('한국어 분석을 시작합니다 재미있어요~~'))

# 결과 : ['한국어 분석을 시작합니다', '재미있어요~~']
```

> 한 문장씩 나눠주는 함수

```
print(kkma.nouns('한국어 분석을 시작합니다 재미있어요~~'))

# 결과 : ['한국어', '분석']
```

> 문장의 단어만 뽑아주는 함수.

```
print(kkma.pos('한국어 분석을 시작합니다 재미있어요~~'))

# 결과
# [('한국어', 'NNG'), ('분석', 'NNG'), ('을', 'JKO'), ('시작하', 'VV'), ('ㅂ니
# ('재미있', 'VA'), ('어요', 'EFN'), ('~~', 'SW')]
```

> 해당 단어가 품사가 명사인지 부사인지 같이 알려주는 함수 (품사 정보는 카페 참조)

Hannanum 한나눔

```
from konlpy.tag import Hannanum
hannanum = Hannanum()

print(hannanum.nouns('한국어 분석을 시작합니다 재미있어요~~'))

# 결과 : ['한국어', '분석', '시작']
```

> 문장의 단어만 뽑아주는 함수.

```
print(hannanum.morphs('한국어 분석을 시작합니다 재미있어요~~'))

# 결과 : ['한국어', '분석', '을', '시작', '합', '니다', '재미있', '어요', '~~']
```

```
print(hannanum.pos('한국어 분석을 시작합니다 재미있어요~~'))

# [('한국어', 'N'), ('분석', 'N'), ('을', 'J'), ('시작', 'N'), ('하', 'X'), (
# ('재미있', 'P'), ('어요', 'E'), ('~~', 'S')]
```

> 해당 단어가 품사가 명사인지 부사인지 같이 알려주는 함수

Twitter

```
from konlpy.tag import Okt

t = Okt()
```

> 과거에는 Okt 가 아니고 twitter 이었다. 나중에 과거버전인지 확인할것

```
print(t.nouns('한국어 분석을 시작합니다 재미있어요~~'))
# 결과 : ['한국어', '분석', '시작']

print(t.morphs('한국어 분석을 시작합니다 재미있어요~~'))
# 결과 : ['한국어', '분석', '을', '시작', '합니다', '재미있어요', '~~']
```

```
print(t.pos('한국어 분석을 시작합니다 재미있어요~~'))
# 결과
# [('한국어', 'Noun'), ('분석', 'Noun'), ('을', 'Josa'), ('시작', 'Noun'),
# ('합니다', 'Verb'), ('재미있어요', 'Adjective'), ('~~', 'Punctuation')]
```

Komoran 코모란

```
from konlpy.tag import Komoran
k = Komoran()
```

```
print(k.nouns('한국어 분석을 시작합니다 재미있어요~~'))
# 결과 : ['한국어', '분석', '시작']

print(k.morphs('한국어 분석을 시작합니다 재미있어요~~'))
# 결과 : ['한국어', '분석', '을', '시작', '하', 'ㅂ니다', '재미있', '어요', '~',

print(k.pos('한국어 분석을 시작합니다 재미있어요~~'))
# 결과 : [('한국어', 'NNP'), ('분석', 'NNG'), ('을', 'JKO'), ('시작', 'NNG'),
# ('ㅂ니다', 'EC'), ('재미있', 'VA'), ('어요', 'EC'), ('~', 'SO'), ('~', 'SO'
```

응용

```
sentence = u'감정노동자 보호법은 사업주로 하여금 감정노동으로부터 근로자를 보호하는
이행하도록 강제한다.\
다만 현장 근로자들을 중심으론 이 같은 법안이 현장에 제대로 적용되기 위해서는 회사의 수
구조와 인력 부족 문제 등\
구조적 문제가 우선 해결돼야 한다는 지적도 나온다.'

sentences = [sentence] * 10000

import time

morphs_processors = [('Hannanum', Hannanum()),
                     ('Kkma', Kkma()),
                     ('Komoran', Komoran()),
                     ('Okt', Okt())]
```

```
for name, morphs_processor in morphs_processors:
    start_time = time.time()
    morphs = [morphs_processor.morphs(sentence) for sentence in sentences]
    elapsed_time = time.time() - start_time
    print('morphs_processor name = %20s, %.5f secs' % (name, elapsed_time))
```

빠른속도와 보통의 정확도를 원한다면 “komoran” 또는 “Hannanum”

속도는 느리더라도 정확하고 상세한 품사 정보를 원한다면 “Kkma”

어느정도의 띄어쓰기 되어있는 “인터넷” 영화평/상품명을 처리할 땐 “Okt”
(만약 띄어쓰기가 없다면 느린 처리속도는 감수해야 함)

언어를 분석할 때, stopwords 라는 용어가 나온다.
stopwords 또는 불용어 란, 우리가 언어를 분석할 때,
의미가 있는 단어와, 의미가 없는 단어나 조사 등이 있다.

이렇게 의미가 없는 것들을 stopwords 라고 한다.

예를 들어서, 다음 문장이 있으면,
“Family is not an important thing. It’s everything.”

Family, important, thing, everything은 의미가 있다고 보고,
나머지 아래 같은 것들은 의미가 없다고 판단하여
stopwords로 정의 한다.

Alice, Storm Troofer 워드클라우드 만들기

Alice

```
# 워드 클라우드
# WordCloud 설치 : pip install wordcloud

from wordcloud import WordCloud, STOPWORDS
```

```
import numpy as np
from PIL import Image
```

```
# 워드 클라우드 폰트 설정
import matplotlib.pyplot as plt
import platform
path = "c:/Windows/Fonts/malgun.ttf"
from matplotlib import font_manager, rc

if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
elif platform.system() == 'Windows':
    path = "c:/Windows/Fonts/malgun.ttf"
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system.... sorry')
```

> 윈도우와 맥용으로 나뉘며 자주 사용하는 코드이다.

```
# alice.txt
text = open('./data/09. alice.txt').read()
alice_mask = np.array(Image.open('./data/09. alice_mask.png'))

stopwords = set(STOPWORDS)
stopwords.add("said")

plt.figure(figsize=(8,8))
plt.imshow(alice_mask, cmap=plt.cm.gray, interpolation='bilinear')
plt.axis('off')
plt.show()
```

```
# 워드클라우드 준비
wc = WordCloud(background_color='white',
               max_words=2000,
               mask=alice_mask,
               stopwords=stopwords)
wc = wc.generate(text)
```

```
# wc.words_
plt.figure(figsize=(12, 12))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Storm Troofer

```
# a_new_hope.txt
text = open('./data/09. a_new_hope.txt').read()

mask = np.array(Image.open('./data/09. stormtrooper_mask.png'))

stopwords = set(STOPWORDS)
stopwords.add("int")
stopwords.add("ext")

wc = WordCloud(max_words=1000,
               mask=mask,
               stopwords=stopwords,
               margin=10,
               random_state=1).generate(text)

default_colors = wc.to_array()

import random

def grey_color_func(word, font_size, position, orientation, random_state=No
    return 'hsl(0, 0%%, %d%%)' % random.randint(60, 100)

plt.figure(figsize=(12, 12))

plt.imshow(wc.recolor(color_func=grey_color_func, random_state=3),
           interpolation='bilinear')
plt.axis('off')
plt.show()
```