**Konlpy를 활용한 한글 텍스트 분석 및 시각화**

텍스트 분석 순서

|  |
| --- |
| 문장(문장을 공백 단위로 분할하여 단어로 구성)  ->단어  ->키워드(사전과 대입하여 키워드 추출)  ->필터링(필요 없는 단어 필터링)  ->집계(결과 집계)  ->시각화(워드클라우드의 시각화) |

코딩 과정

|  |  |
| --- | --- |
| 파이썬 패키지 설치 방법  python -m pip install 패키지 이름  현재 패키지가 무엇이 있는지 확인하고 필요한 모듈을 설치해준다.  현재 패키지 보는법 : pip list |  |
| 한글 분석에 특화된 패키지 설치하기  !pip install konlpy  워드클라우드 설치하기  !pip install wordcloud  현재 패키지 리스트를 보면 설치된걸 확인할 수 있음. |  |
| 파일을 불러올 때 경로를 /대신 \로 표시하여 경로로 표시한 부분이 유니코드로 인식되어 발생한 에러다. 또는 \\ 써주면 된다. 하지만 그래도 에러 발생했다.    환경변수를 설정해도 에러가 발생해서 직접 변경해주었다. | |

코드 1

|  |
| --- |
| #Step 1. 필요한 모듈을 실행합니다  from konlpy.tag import Kkma #한글 분석에 특화된 패키지  import matplotlib.pyplot as plt  from matplotlib import font\_manager, rc  from wordcloud import WordCloud  # wordcloud 설치할 때 setuptools 에러나면 아래 코드 실행후 재실행  # pip install -U pip setuptools  from collections import Counter #빈도를 찾기 위한 모듈  import numpy as np  kkma = Kkma( )  #앞은 대문자 뒤는 소문자k, 문자을 입력받으면 형태소 분석 후 명사를 골라내줌  #단어가 형용사인지 명사인지 Kkma함수 사용  # Step 2. 파일 불러와서 형태소 분석하기  data1 = open("c:\\temp\\경주여행\_지식인\_2016\_2.txt").read()  #웹페이지 크롤링을 해서 데이터를 읽어옴  data1  data2 = kkma.nouns(data1) #nouns는 명사만 골라와라라는 뜻  data3 = Counter(data2) #명사만 골라서 빈도를 나타냄  # step 3. 불용어 제거하기  #파일안에 불필요한 단어를 적어놓고 모두 없어질 때까지 for문 반복으로 일괄 삭제  stop\_words=open(”c:\\temp\\stop\_words.txt”).read()  data3=[each\_word for each\_word in data2  if each\_word not in stop\_words]  #불용어 제거 방법 1 : 메모장에 불용어 정리해서 for  #불용어 제거 방법 2 : 글자 수 지정해서 없애기( 밑에 내용 확인하기)  data3  #1글자 이하이거나 10글자 이상인 단어 삭제하기  #인터넷은 띄어쓰기가 안되어 있으면 한단어로 취급하기 떄문에 글자 제한수를 줘야함.  data4=[ ]  for i in range(0,len(data3)):  if len(data3[i]>=2 | len(data3[i]<=10:  data4.append(data3[i])  #Step 4. 단어 별 빈도수 집계하기  data5 = Counter(data4)  data6 = data5.most\_common(100) #가장 많이 언급된 단어보기위한 함수 most\_common  tmp\_data = dict(data6)  # Step 5. 워드 클라우드 그리기  #워드 클라우드는 언급 빈도수에 따라 글자 크기를 설정  #워드 클라우드는 기본적으로 한글폰트를 지원되지 않음.  #font\_path를 이용해서 경로를 써준다.  wordcloud = WordCloud(font\_path="C:/Users/wjdtn/Downloads/나눔 글꼴\나눔바른고딕옛한글/NanumBarunGothic-YetHangul.otf"  ,relative\_scaling = 0.2, #크기  background\_color='black',  ).generate\_from\_frequencies(tmp\_data)  plt.figure(figsize=(8,4))  plt.imshow(wordcloud)  plt.axis('on')  plt.show( ) |

코드 2

|  |
| --- |
| #가져온 그림 모양대로 워드클라우드 표현하기, 마스킹한다고 표현함.  import numpy as np #pip install numpy 해야 함  from PIL import Image #pip install Image 해야 함  from wordcloud import ImageColorGenerator  alice\_mask = np.array(Image.open("C:/Users/wjdtn/Downloads/alice.jpg"))  #해당 사진의 모양대로 워드클라우드를 그리는 것  wc = WordCloud(  font\_path="C:/Users/wjdtn/Downloads/나눔글꼴\나눔바른고딕옛한글/NanumBarunGothic-YetHangul.otf" ,  relative\_scaling = 0.2, mask=alice\_mask , #mask : 그림을 그리는 작업  background\_color='white',  min\_font\_size=1 ,  max\_font\_size=40,  max\_words=2000,  ).generate\_from\_frequencies(tmp\_data)  plt.figure(figsize=(8,8))  plt.imshow(wc)  plt.axis("off")  plt.show( ) |

코드 3

|  |
| --- |
| #워드 클라우드를 그래프로 나타내기  # 그래프에 출력되는 한글을 위해 폰트 설정  import matplotlib.font\_manager as fm  import matplotlib.pyplot as plt  font\_location = "C:/Users/wjdtn/Downloads/나눔 글꼴\나눔바른고딕옛한글/NanumBarunGothic-YetHangul.otf"  font\_name = fm.FontProperties(fname = font\_location).get\_name()  matplotlib.rc('font', family=font\_name)  import nltk #빈도를 나타내주는 모듈  plt.figure(figsize=(20,4))  from nltk.probability import FreqDist  g\_data4 = FreqDist(data4)  #g\_data4 = nltk.Text(data4 , name=" 다빈도 단어 그래프 출력하기")  g\_data4.plot(50) |