**NItk를 활용한 영어 텍스트 분석 및 시각화**

- 영어텍스트 분석 -> 워드 클라우드 시각화 ->자주 언급되는 단어의 그래프화

- 한글 텍스트는 Konlpy, 영어 텍스트는 Nltk 패키지를 사용한다.

코딩 과정

|  |  |
| --- | --- |
| nltk.download()를 실행하면 위와 같은 링크가 나오고 패키지 모두 다운받기 | |
| 1. word\_tokenize()를 사용하면 단어별로만 짜르기때문에 don't의 경우 do와 n't으로 분류된다. 공백이나, ','를 기준으로 단어별로 짜름. 하지만 축약형단어를 제대로 분류할 수 없기 때문에 WordPunctTokenizer를 사용 추천.    때문에 2. WordPunctTokenizer를 사용하면 don't의 경우 don과 t로 분류된다.    3. 정규식을 이용해서 don't를 하나의 단어처럼 나오도록 하려면 아래 코드를 필수로 써야함.   |  | | --- | | from nltk.tokenize.regexp import RegexpTokenizer  tokenizer = RegexpTokenizer("[\w']+")  new\_data3 = tokenizer.tokenize(data1)  print(new\_data3) | | |
| #data3 = nltk.pos\_tag(data2)  # 추출된 단어를 품사별로 더 자세하기 확인하기  #data3  위의 코드를 실행하면 단어별 품사도 알 수 있다. |  |
| stopwords.words()는 NLTK가 정의한 영어 불용어 리스트를 의미한다. |  |
| nltk.Text()안에는 그래프 그리는 기능이 존재한다. | |

코드

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # 파이썬을 활용한 영어 텍스트 분석  # nltk 패키지 사용  # cmd 창에서 pip install nltk 한 후 아래 코드 진행  # 참고 사이트-http://excelsior-cjh.tistory.com/63?category=928322  import nltk  #nltk.download()  # nltk 패키지를 사용하기 위한 데이터를 다운로드 하기 - 시간이 몇 분 걸림  #nltk 패키지가 정상적으로 설치 되었는 지 아래의 방법으로 테스트하기  from nltk.corpus import brown  brown.words() #다운로드가 잘 됐으면 단어 몇개가 출력될 것이다.  #불용어 제거를 위한 불용어 사전 로딩  from nltk.corpus import stopwords  from nltk.tokenize import RegexpTokenizer  # 분석할 파일을 불러옴  data1 = ''' i love apple , apple , apple and orange , orange and banana.  i don't love peach '''  #영어의 경우 축약형이 많기 때문에 중요하다.  data1   |  | | --- | |  |   # 토큰화 하기 - 단어별로 짤라내기  #wrod\_tokenize : 일정한 기준으로 단어를 분할하는 명령이지만 축약형 단어를 제대로 분류할 수 없음.  data2 = nltk.word\_tokenize(data1)  print(data2)  from nltk.tokenize import WordPunctTokenizer #이용추천  tokenizer = WordPunctTokenizer()  new\_data2 = tokenizer.tokenize(data1)  print(new\_data2)  #축약형의 경우 축약된 단어를 분리하지 말고 출력하고 싶을 경우 정규식을 사용하면 됨  from nltk.tokenize.regexp import RegexpTokenizer  tokenizer = RegexpTokenizer("[\w']+")  new\_data3 = tokenizer.tokenize(data1)  print(new\_data3)  # 추출된 단어별 상세 품사를 알고 싶다면 아래 코드 실행  #data3 = nltk.pos\_tag(data2)  # 추출된 단어를 품사별로 더 자세하기 확인하기  #data3   |  | | --- | |  |   # 불용어 제거하기  data4 = [ each\_word for each\_word in new\_data3  if each\_word not in stopwords.words() ]  data4   |  | | --- | |  |   # 단어별로 언급빈도 집계하기  from collections import Counter  data5 = Counter(data4)  print(data5)  data6 = data5.most\_common(100)  print(data6)  data7 = dict(data6)  print(data7)    # 집계된 단어를 그래프로 표시하기  g\_data4 = nltk.Text(data4 , name=" 다빈도 단어 그래프 출력하기")  #nltk안에는 그래프그리는 기능이 있음  g\_data4.plot(10)    # 워드 클라우드 그리기  from wordcloud import WordCloud  import matplotlib.pyplot as plt  wordcloud = WordCloud(font\_path='c:\windows\fonts\HMKMG.TTF' ,  relative\_scaling = 0.2,  background\_color='black',  ).generate\_from\_frequencies(data7)  plt.figure(figsize=(10,4))  plt.imshow(wordcloud)  plt.axis('on')  plt.show( )    # 스탠포드 대학교 졸업식에서 연설한 스티브 잡스의 연설문 분석하기  # 파일 불러와서 형태소 분석하기  data1 = open("c:\\temp\\steve.txt").read()  # word\_tokenize 함수는 축약형을 제대로 분리할 수 없음  # 그래서 아래의 방법을 추천  from nltk.tokenize import WordPunctTokenizer  tokenizer = WordPunctTokenizer()  new\_data2 = tokenizer.tokenize(data1)  print(new\_data2)  # 축약형의 경우 축약된 단어를 분리하지 말고 출력하고 싶을 경우 정규식을 사용하면 됨  from nltk.tokenize.regexp import RegexpTokenizer  tokenizer = RegexpTokenizer("[\w']+")  new\_data3 = tokenizer.tokenize(data1)  print(new\_data3)   |  | | --- | |  |   # 불용어 제거하기  data4 = [ each\_word for each\_word in new\_data3  if each\_word not in stopwords.words() ]  data4   |  | | --- | |  |   # 단어별로 언급빈도 집계하기  from collections import Counter  data5 = Counter(data4)  #print(data5)  data6 = data5.most\_common(100)  #print(data6)  data7 = dict(data6)  #print(data7)  # 집계된 단어를 그래프로 표시하기  plt.figure(figsize=(10,4))  g\_data4 = nltk.Text(data4 , name=" 다빈도 단어 그래프 출력하기")  g\_data4.plot(20)    # 워드 클라우드 그리기  from wordcloud import WordCloud  import matplotlib.pyplot as plt  wordcloud = WordCloud(font\_path='c:\windows\fonts\HMKMG.TTF' ,  relative\_scaling = 0.2,  background\_color='black',  ).generate\_from\_frequencies(data7)  plt.figure(figsize=(10,4))  plt.imshow(wordcloud)  plt.axis('on')  plt.show( )    # 추가로 불용어 제거한 후 워드 클라우드 그리기  stop\_words = open("c:\\temp\\eng\_stop\_word.txt").read()  new\_data4 = [ each\_word for each\_word in data4  if each\_word not in stop\_words ]  new\_data4  #1글자 이하이거나 10 글자 이상인 단어 삭제하기  new\_data5=[ ]  for i in range(0,len(new\_data4)) :  if len(new\_data4[i]) >= 2 | len(new\_data4[i]) <= 10 :  new\_data5.append(new\_data4[i])  data5 = Counter(new\_data5)  data6 = data5.most\_common(100)  data7 = dict(data6)  wordcloud = WordCloud(font\_path='c:\windows\fonts\HMKMG.TTF' ,  relative\_scaling = 0.2,  background\_color='black',  ).generate\_from\_frequencies(data7)  plt.figure(figsize=(10,4))  plt.imshow(wordcloud)  plt.axis('on')  plt.show( )    # 스티브 잡스 얼굴 사진 가져와서 워드 클라우드 그리기  import numpy as np  from PIL import Image #pip install Image 해야 함  from wordcloud import ImageColorGenerator  steve\_mask = np.array(Image.open("C:\\python 3.5\\py\_test\\steve.jpg"))  wc = WordCloud(font\_path='c:\windows\fonts\HMKMG.TTF' ,  relative\_scaling = 0.4, mask=steve\_mask ,  background\_color='white',  min\_font\_size=1 ,  max\_font\_size=60,  max\_words=2000,  ).generate\_from\_frequencies(data7)  plt.figure(figsize=(8,8))  plt.imshow(wc)  plt.axis("off")  plt.show( )    plt.figure(figsize=(10,4))  g\_data5 = nltk.Text(new\_data5 , name=" 다빈도 단어 그래프 출력하기")  g\_data5.plot(50) |