Colab 환경에서 자연어처리(2)

colab에서 열때,

학습 내용

01 사전 작업

02 텍스트 데이터 시각화

03 영화 댓글 시각화

01 사전 작업

- 한글 폰트 적용, konlpy 설치

In [1]:

```
### 나눔 고딕 설치
!apt-get update -qq   # 설치를 업데이트
!apt-get install fonts-nanum* -qq  # 설치한다. fonts-nanum*
```

```
Selecting previously unselected package fonts-nanum.
(Reading database ... 160772 files and directories currently installed.)
Preparing to unpack .../fonts-nanum_20170925-1_all.deb ...
Unpacking fonts-nanum (20170925-1) ...
Selecting previously unselected package fonts-nanum-eco.
Preparing to unpack .../fonts-nanum-eco_1.000-6_all.deb ...
Unpacking fonts-nanum-eco (1.000-6) ...
Selecting previously unselected package fonts-nanum-extra.
Preparing to unpack .../fonts-nanum-extra_20170925-1_all.deb ...
Unpacking fonts-nanum-extra (20170925-1) ...
Selecting previously unselected package fonts-nanum-coding.
Preparing to unpack .../fonts-nanum-coding_2.5-1_all.deb ...
Unpacking fonts-nanum-coding (2.5-1) ...
Setting up fonts-nanum-extra (20170925-1) ...
Setting up fonts-nanum (20170925-1) ...
Setting up fonts-nanum-coding (2.5-1) ...
Setting up fonts-nanum-eco (1.000-6) ...
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
```

In [4]:

```
import matplotlib.font_manager as fm # 폰트 관련 용도
import matplotlib.pyplot as plt      # 그래프 그리는 용도

path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf' # 설치된 나눔 글꼴중 원하는 녀석의 전체
font_name = fm.FontProperties(fname=path, size=10).get_name()
print(font_name)
plt.rc('font', family=font_name)

# 우선 fm._rebuild() 를 해주고 # 폰트 매니저 재빌드가 필요하다.
fm._rebuild()
```

```
NanumGothic Eco
```

런타임 재기동 후, 다시 시작

In [3]:

```
%matplotlib inline
import matplotlib as mpl # 기본 설정 만지는 용도
import matplotlib.pyplot as plt # 그래프 그리는 용도
import matplotlib.font_manager as fm # 폰트 관련 용도
import numpy as np

path = '/usr/share/fonts/truetype/nanum/NanumGothicEco.ttf' # 설치된 나눔글꼴중 원하는 녀석의 전체
font_name = fm.FontProperties(fname=path, size=10).get_name()
print(font_name)
plt.rc('font', family=font_name)

## 음수 표시되도록 설정
mpl.rcParams['axes.unicode_minus'] = False
```

```
NanumGothic Eco
```

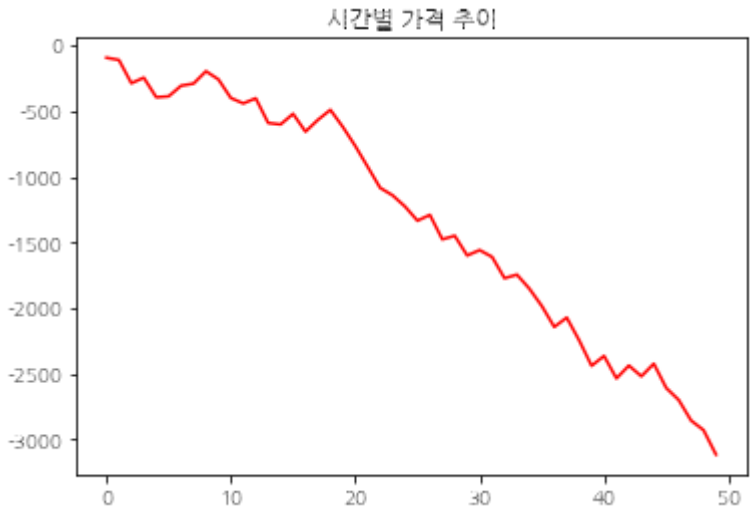
In [4]:

```
# 데이터 준비
data = np.random.randint(-200, 100, 50).cumsum()

# 그래프를 그려 한글 확인
plt.plot(range(50), data, 'r')
plt.title('시간별 가격 추이')
```

Out[4]:

```
Text(0.5, 1.0, '시간별 가격 추이')
```



웹 환경이 아닌 개인 컴퓨터에서의 한글 폰트 설정

```
from matplotlib import font_manager, rc
import matplotlib.pyplot as plt
import platform

path = "C:/Windows/Fonts/malgun.ttf"  # 한글 폰트 위치 지정
if platform.system() == "Windows":  # 사용 OS가 Windows의 경우
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":   # 사용 OS가 Mac인 경우
    rc('font', family='AppleGothic')
else:
    print("Unknown System")
```

konlpy 설치

In [5]:

```
pip install konlpy
```

```
Collecting konlpy
  Downloading https://files.pythonhosted.org/packages/85/0e/f385566fec837c0b83f216b2
da65db9997b35dd675e107752005b7d392b1/konlpy-0.5.2-py2.py3-none-any.whl (https://file
s.pythonhosted.org/packages/85/0e/f385566fec837c0b83f216b2da65db9997b35dd675e1077520
05b7d392b1/konlpy-0.5.2-py2.py3-none-any.whl) (19.4MB)
     |████████████████████████████████| 19.4MB 1.3MB/s
Collecting beautifulsoup4==4.6.0
  Downloading https://files.pythonhosted.org/packages/9e/d4/10f46e5cfac773e22707237b
fcd51bbffeaf0a576b0a847ec7ab15bd7ace/beautifulsoup4-4.6.0-py3-none-any.whl (https://
files.pythonhosted.org/packages/9e/d4/10f46e5cfac773e22707237bfcd51bbffeaf0a576b0a84
7ec7ab15bd7ace/beautifulsoup4-4.6.0-py3-none-any.whl) (86kB)
     |████████████████████████████████| 92kB 9.8MB/s
Collecting JPype1>=0.7.0
  Downloading https://files.pythonhosted.org/packages/98/88/f817ef1af6f794e8f11313dc
d1549de833f4599abcec82746ab5ed086686/JPype1-1.3.0-cp37-cp37m-manylinux_2_5_x86_64.ma
nylinux1_x86_64.whl (https://files.pythonhosted.org/packages/98/88/f817ef1af6f794e8f
11313dcd1549de833f4599abcec82746ab5ed086686/JPype1-1.3.0-cp37-cp37m-manylinux_2_5_x8
6_64.manylinux1_x86_64.whl) (448kB)
     |████████████████████████████████| 450kB 32.5MB/s
Requirement already satisfied: tweepy>=3.7.0 in /usr/local/lib/python3.7/dist-packag
es (from konlpy) (3.10.0)
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.7/dist-packages
 (from konlpy) (1.19.5)
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.7/dist-packages
(from konlpy) (4.2.6)
Collecting colorama
  Downloading https://files.pythonhosted.org/packages/44/98/5b86278fbbf250d239ae0ecb
724f8572af1c91f4a11edf4d36a206189440/colorama-0.4.4-py2.py3-none-any.whl (https://fi
les.pythonhosted.org/packages/44/98/5b86278fbbf250d239ae0ecb724f8572af1c91f4a11edf4d
36a206189440/colorama-0.4.4-py2.py3-none-any.whl)
Requirement already satisfied: typing-extensions; python_version < "3.8" in /usr/loc
al/lib/python3.7/dist-packages (from JPype1>=0.7.0->konlpy) (3.7.4.3)
Requirement already satisfied: requests[socks]>=2.11.1 in /usr/local/lib/python3.7/d
ist-packages (from tweepy>=3.7.0->konlpy) (2.23.0)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages
(from tweepy>=3.7.0->konlpy) (1.15.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/
dist-packages (from tweepy>=3.7.0->konlpy) (1.3.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/loca
l/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy>=3.7.0->konlpy)
 (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-p
ackages (from requests[socks]>=2.11.1->tweepy>=3.7.0->konlpy) (2021.5.30)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-package
s (from requests[socks]>=2.11.1->tweepy>=3.7.0->konlpy) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pa
ckages (from requests[socks]>=2.11.1->tweepy>=3.7.0->konlpy) (3.0.4)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6; extra == "socks" in /usr/loca
l/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy>=3.7.0->konlpy)
 (1.7.1)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-pack
ages (from requests-oauthlib>=0.7.0->tweepy>=3.7.0->konlpy) (3.1.1)
Installing collected packages: beautifulsoup4, JPype1, colorama, konlpy
  Found existing installation: beautifulsoup4 4.6.3
    Uninstalling beautifulsoup4-4.6.3:
```

```
      Successfully uninstalled beautifulsoup4-4.6.3
Successfully installed JPype1-1.3.0 beautifulsoup4-4.6.0 colorama-0.4.4 konlpy-0.5.2
```

In [6]:

```
import nltk
from konlpy.tag import Kkma      ### 꼬꼬마
from konlpy.tag import Hannanum  ### 한나눔
```

In [7]:

```
### wordcloud와 이미지 표시
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
```

02 텍스트 데이터 시각화

- open("불러올 파일명").read() : 파일 내용을 불러온다.

In [8]:

```
### 데이터 읽기
text = open("alice.txt").read()
text
```

Out[8]:

```
'WufeffProject GutenbergW's AliceW's Adventures in Wonderland, by Lewis CarrollWnWnT
his eBook is for the use of anyone anywhere at no cost and withWnalmost no restricti
ons whatsoever.  You may copy it, give it away orWnre-use it under the terms of the
Project Gutenberg License includedWnwith this eBook or online at www.gutenberg.orgWn
WnWnTitle: AliceW's Adventures in WonderlandWnWnAuthor: Lewis CarrollWnWnPosting Dat
e: June 25, 2008 [EBook #11]WnRelease Date: March, 1994Wn[Last updated: December 20,
2011]WnWnLanguage: EnglishWnWnWn*** START OF THIS PROJECT GUTENBERG EBOOK ALICEW'S A
DVENTURES IN WONDERLAND ***WnWnWnWnWnWnWnWnWnWnWnALICEW'S ADVENTURES IN WONDERLANDWn
WnLewis CarrollWnWnTHE MILLENNIUM FULCRUM EDITION 3.0WnWnWnWnWnCHAPTER I. Down the R
abbit-HoleWnWnAlice was beginning to get very tired of sitting by her sister on the
Wnbank, and of having nothing to do: once or twice she had peeped into theWnbook her
sister was reading, but it had no pictures or conversations inWnit, W'and what is th
e use of a book,W' thought Alice W'without pictures orWnconversation?W'WnWnSo she wa
s considering in her own mind (as well as she could, for theWnhot day made her feel
very sleepy and stupid), whether the pleasureWnof making a daisy-chain would be wort
h the trouble of getting up andWnpicking the daisies, when suddenly a White Rabbit w
ith pink eyes ranWnclose by her.WnWnThere was nothing so VERY remarkable in that; no
```

In [9]:

```
### wordcloud의 불용어 단어 확인
print( type(STOPWORDS) )
print(STOPWORDS)
```

```
<class 'set'>
{'because', 'so', 'up', "haven't", "she'll", 'against', "aren't", 'other', 'to', "we
ren't", 'they', 'this', 'hers', "you're", 'yours', 'having', 'themselves', 'who', "d
idn't", "what's", "isn't", 'myself', 'whom', "they'd", 'at', 'below', "you'll", 'thr
ough', 'were', 'own', 'each', 'http', 'of', 'nor', "let's", 'their', 'yourself', 'an
d', 'shall', 'what', 'ourselves', 'out', 'that', "it's", 'www', 'both', 'very', 'doe
s', 'him', 'which', 'any', "couldn't", 'his', 'on', 'you', 'from', "hadn't", "was
n't", 'by', 'been', 'its', 'her', 'after', "i'm", 'like', 'as', 'r', "we're", "ca
n't", 'how', 'could', 'else', 'down', 'when', 'can', 'are', "where's", 'was', "shoul
dn't", 'com', 'only', 'here', "who's", 'all', 'if', "hasn't", 'into', 'more', 'am',
'doing', "he's", "that's", 'why', 'once', "there's", 'there', 'for', 'cannot', 'bu
t', "i've", 'me', 'too', 'again', 'during', "he'll", "wouldn't", 'them', "why's", 's
he', 'these', 'just', 'a', 'yourselves', 'he', 'where', "won't", "she'd", 'also', 't
he', 'an', 'get', 'my', 'no', 'i', "shan't", 'ever', "they've", 'than', 'had', 'sinc
e', 'theirs', 'however', 'do', 'has', 'in', "i'll", 'until', "i'd", 'be', 'ought',
'some', 'would', 'your', 'otherwise', 'should', 'such', 'did', 'under', "they'll",
'while', 'before', 'between', 'herself', 'not', "when's", "they're", 'being', 'ove
r', 'with', 'our', "he'd", "we've", "don't", 'is', 'ours', "you'd", "how's", 'sh
e's", "doesn't", 'himself', "you've", 'k', "mustn't", "we'd", 'same', 'most', 'furth
er', 'about', 'few', 'then', 'those', "we'll", 'have', 'above', "here's", 'we', 'i
t', 'or', 'off', 'itself'}
```

파이썬 자료형 집합에 대해 알아보기

- 중복을 없앤다.

In [10]:

```
## 집합 확인
s2 = set([1,2,3,4,5,1,2])
s2
```

Out[10]:

```
{1, 2, 3, 4, 5}
```

In [13]:

```
print( len(STOPWORDS) )
stopwords = set(STOPWORDS)
print( len(stopwords) )
```

```
190
190
```

In [14]:

```
### 불용어 단어 추가
stopwords = set(STOPWORDS)
stopwords.add("said")
print( len(stopwords) )  # 190 -> 191로 변경
stopwords
```

191

앨리스 이미지 확인

- Image.open() : 주어진 이미지 파일을 불러온다.
- np.array() : 배열을 만든다

In [15]:

```
# alice_mask = np.array(Image.open("alice_color.png"))
alice_mask = np.array(Image.open("alice_color.png"))
alice_mask[0]   # 이미지 배열화된 내용 하나의 정보 확인
```

Out[15]:

```
array([[255, 255, 255, 255],
       [255, 255, 255, 255],
       [255, 255, 255, 255],
       ...,
       [255, 255, 255, 255],
       [255, 255, 255, 255],
       [255, 255, 255, 255]], dtype=uint8)
```

In [16]:

```
### 워드 클라우드 표현을 위한 데이터 생성
wc = WordCloud( background_color='white',    # 배경색
                max_words=2000,              # 최대 표시 단어
                mask=alice_mask,             # 마스크 이미지
                contour_width=3,             # 외곽선
                contour_color="steelblue" )  # 색
wc.generate(text)
wc.words_
```

Out[16]:

```
{'Alice': 1.0,
 'said': 0.8353909465020576,
 'said Alice': 0.5061728395061729,
 'little': 0.4444444444444444,
 'one': 0.39094650205761317,
 'know': 0.37037037037037035,
 'went': 0.34156378600823045,
 'thing': 0.3292181069958848,
 'time': 0.3168724279835391,
 'Queen': 0.31275720164609055,
 'see': 0.2757201646090535,
 'now': 0.24691358024691357,
 'began': 0.23868312757201646,
 'way': 0.2345679012345679,
 'head': 0.23045267489711935,
 'Mock Turtle': 0.23045267489711935,
 'say': 0.22633744855967078,
```

- interpolation 참조 : https://matplotlib.org/3.2.1/gallery/images_contours_and_fields/interpolation_methods.html (https://matplotlib.org/3.2.1/gallery/images_contours_and_fields/interpolation_methods.html)

In [17]:

```
plt.figure(figsize=(15,8))  # 크기
plt.imshow(alice_mask, cmap=plt.cm.gray, interpolation='bilinear')
plt.axis('off')
plt.show()
```

위에서 생성한 wc 데이터를 이용하여 그래프 표시

In [18]:

```
plt.figure(figsize=(15,8))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```

03 영화 댓글 시각화

- 분노의 질주 - 댓글 분석

In [19]:

```
doc_ko = open("15_TheExtreme_utf8.txt").read()
doc_ko[1:1000]
```

Out[19]:

'"x"₩n"1" " 분노의 질주 시리즈중에서 제일 별루"₩n"2" " 스케일 큰 시끄러운 액션이 난무하는데도 이렇게까지 지루할수 있다니....."₩n"3" " 시~원 하게 잘 본 영화. 다음 시리즈에서는 여자 주인공의 비중이 더 높아졌으면 하는 바램!"₩n"4" " 반지닦기, 자살닦이, 고무닦이, 정의닦이...로 이어지는 한심한 DC 시리즈 ₩n레지던트 이블 시리즈 ₩n그리고 이 영화 분노의 질주 시리즈 ₩n공통점은 시리즈가 거듭될수록 돈은 많이 들지만 재미는 없어지고 ₩nCG는 떡질되지만 실감나는 장면은 더 없어지도 뻔히 가짜라는게 드러나는 영화들 ₩n그러나₩n아무리 엉터리로 만들고, 자국에서 망해도 ₩n미국 블록버스터라면 맹목적으로 보는 중국애들 땜에 ₩n아무리 쓰레기 영화라도 본전 건지는 것은 물론 상당히 많은 돈을 버니... ₩n이런 쓰레기들이 매년 양산된다. ₩n물론, 중국애들도 할말은 있을 거다 ₩n공산당이 검열하는 자국영화보다는 낫다고... ₩n하지만 우리들은 다른 전세계의 재미있는 영화를 볼 선택의 자유가 있쟎아! ₩n왜 이런 쓰레기 영화를 보는 거지?"₩n"5" " "₩n"6" " 그냥 액션만 보면 멋진데₩n스토리는 주인공이 전여친한테 싸지른₩n애새끼 구하러 간다며 아빠행세하면서₩n그 덕분에 지동료들 다 버리고 미쳐 날뛰는 내용"₩n"7" " "₩n"8" " 아래는 다들 평점 알바들인가부네.. 이런 개 쓰레기 영화가 평점이 이리 높다니"₩n"9" " "₩n"10" " "₩n"11" " "₩n"12" " "₩n"13" " "₩n"14" " "₩n"15" " 스케일은 점점 더 커지지만, 액션은 멍청할정도로 어이가없음 과유불급"₩n"16" " 이 시리즈로 이렇게 길게 간다는게 신기.. 새로울 건 없지만 달리는 걸 좋아하시는 분이라면 "₩n"17" " "₩n"18" " "₩n"19" " "₩n"20" " 대머리들은 TV물로 찍고,₩n감독은 CG실에서 이어붙히고."₩n"21" " "₩n"22" " "₩n"23" " "₩n"24" " "₩n"25" " "₩n"26" " "₩n"27" " "₩n"28" " "₩n"29" " 이제는 제목'

In [27]:

```
# OKT 클래스를 이용한 명사확인 Okt (11초 : 4286), Kkma (37초 : 5572)
from konlpy.tag import Okt   ### Okt
t = Okt()

#from konlpy.tag import Kkma  ### Okt
#t = Kkma()
doc_nouns = t.nouns(doc_ko)
print( len( doc_nouns ) )
doc_nouns
```

4286

In [28]:

```
# nltk.Text()를 이용하여 nltk가 가지는 많은 기능을 사용 가능해짐.
ko = nltk.Text(doc_nouns, name="분노의 질주")
print(type(ko))
print(len(ko.tokens))
```

```
<class 'nltk.text.Text'>
4286
```

In [36]:

```
### 단어들의 사용 횟수 확인 - 빈도 분석
type( ko.vocab() ), ko.vocab().most_common(20)
```

Out[36]:

```
(nltk.probability.FreqDist,
 [('영화', 157),
  ('액션', 149),
  ('시리즈', 70),
  ('분노', 62),
  ('스토리', 57),
  ('질주', 55),
  ('최고', 44),
  ('자동차', 40),
  ('생각', 37),
  ('편', 37),
  ('더', 36),
  ('그냥', 33),
  ('역시', 32),
  ('좀', 31),
  ('폴', 31),
  ('워커', 30),
  ('그', 28),
  ('장면', 26),
  ('이', 25),
  ('것', 25)])
```

In [37]:

```
most_fre = ko.vocab().most_common(50)
most_fre
```

Out[37]:

```
[('영화', 157),
 ('액션', 149),
 ('시리즈', 70),
 ('분노', 62),
 ('스토리', 57),
 ('질주', 55),
 ('최고', 44),
 ('자동차', 40),
 ('생각', 37),
 ('편', 37),
 ('더', 36),
 ('그냥', 33),
 ('역시', 32),
 ('좀', 31),
 ('폴', 31),
 ('워커', 30),
 ('그', 28),
 ('장면', 26),
 ('이', 25),
 ('것', 25),
 ('가족', 24),
 ('점', 24),
 ('기대', 24),
 ('재미', 23),
 ('내용', 23),
 ('볼', 22),
 ('시간', 22),
 ('보고', 22),
 ('스케일', 21),
 ('진짜', 21),
 ('이번', 21),
 ('볼거리', 20),
 ('정말', 20),
 ('마지막', 19),
 ('액션영화', 19),
 ('돈', 18),
 ('이제', 18),
 ('또', 18),
 ('분노의질주', 17),
 ('사람', 16),
 ('다음', 15),
 ('평점', 15),
 ('정도', 15),
 ('대박', 15),
 ('말', 15),
 ('때', 15),
 ('중간', 15),
 ('전편', 15),
 ('눈', 14),
 ('스트레스', 14)]
```

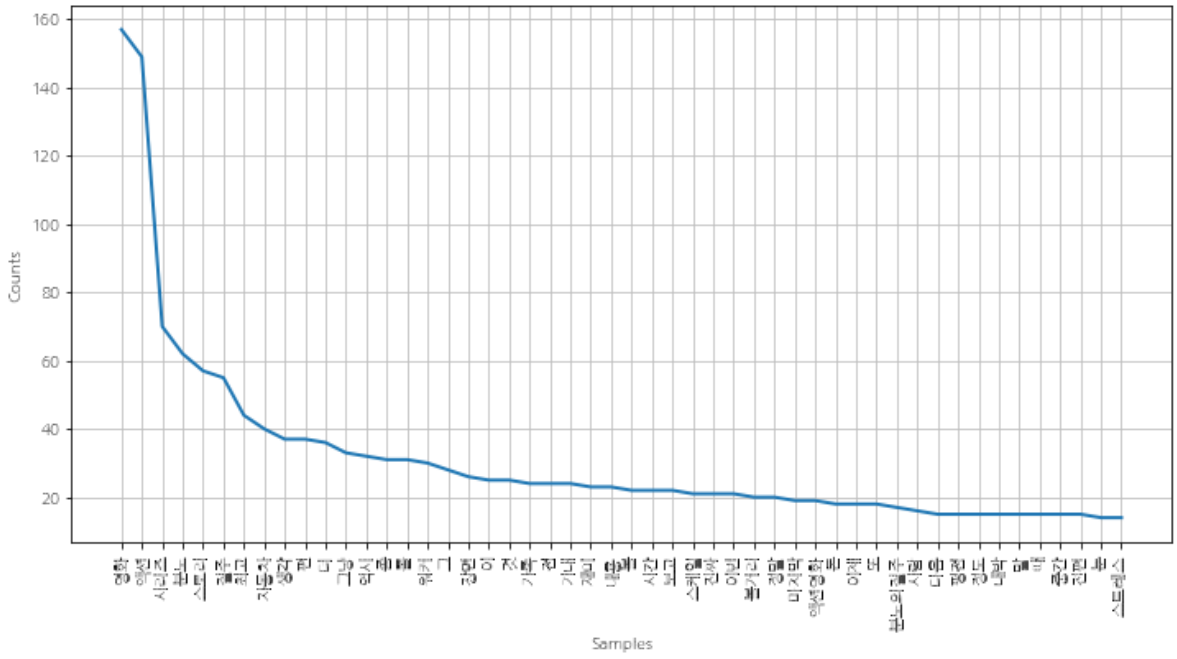
In [38]:

```
### 중복된 단어를 제거한 개수를 확인
print(len(set(ko.tokens)))
```

1381

In [39]:

```
plt.figure(figsize=(12, 6))
ko.plot(50)
plt.show()
```



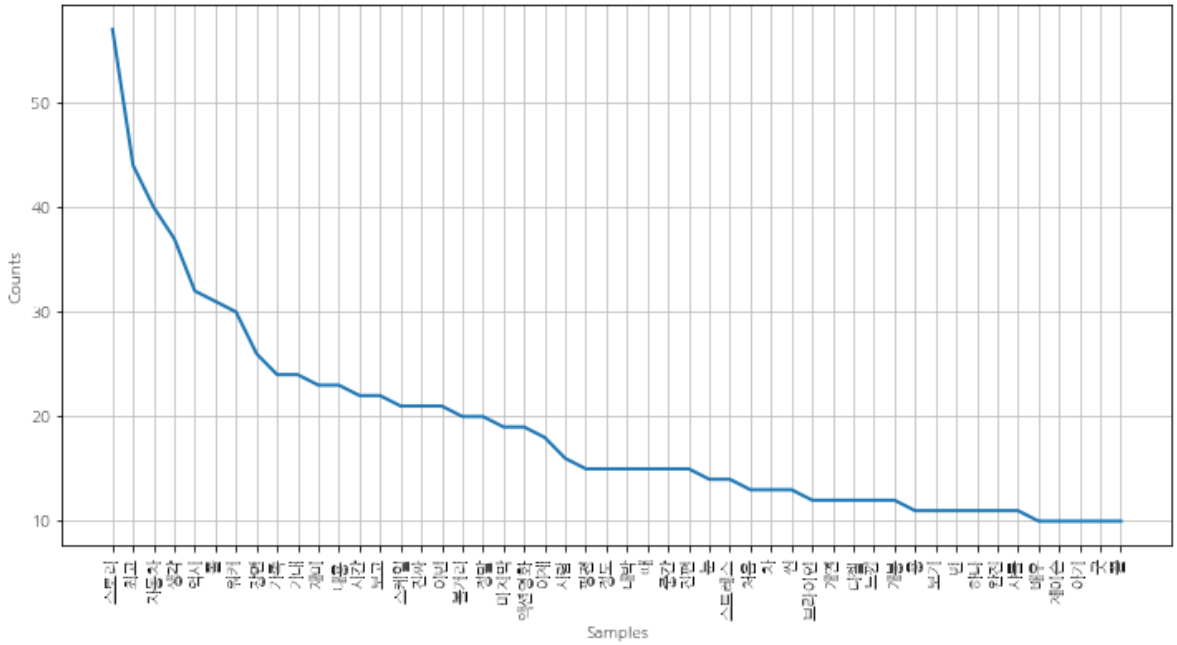
In [40]:

```
### 한글에서는 따로 불용어 사전이 없어, 따로 만들거나 또는 파일로 부터 불러올 수 있다.
stop_words = ['분노', '영화', '액션', '시리즈', '더',
              '그', '이', '것', '또', '좀',
              '돈', '것', '다음', '질주', '그냥',
              '분노의질주', '말', '뭐', '애', '나', '듯', '편', '볼', '점', '중', '로']

new_ko = [ ]
for one_word in ko:
  if one_word not in stop_words:
    new_ko.append(one_word)
```

In [41]:

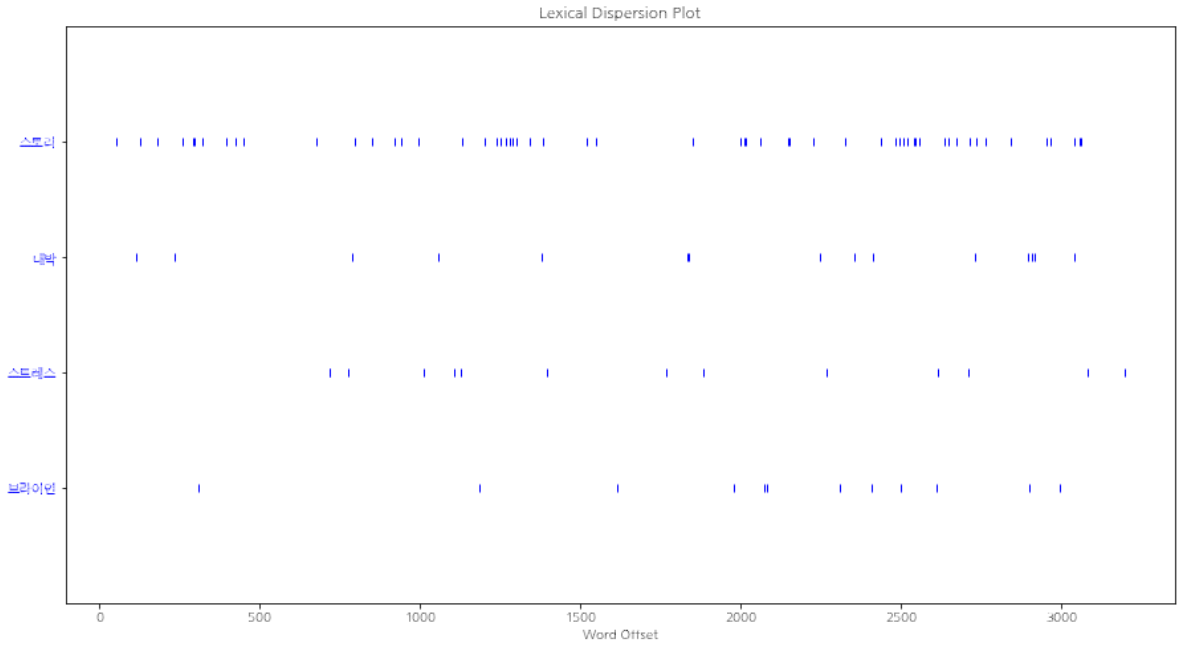
```
### nltk Text 객체 만들기
new_ko = nltk.Text(new_ko, name="분노의 질주2")
plt.figure(figsize=(12,6))
new_ko.plot(50)
```



텍스트의 단어어 분포 확인 (dispersion_plot)

In [42]:

```
plt.figure(figsize=(15,8))
new_ko.dispersion_plot(['스토리', '대박', '스트레스', '브라이언'])
```



In [43]:

```
from wordcloud import WordCloud, STOPWORDS

import numpy as np
from PIL import Image
```

In [44]:

```
Car_mask = np.array(Image.open("Draw_car1.png"))
```

In [45]:

```
data = new_ko.vocab().most_common(1000)
```

In [46]:

```
### 워드 클라우드 표현을 위한 데이터 생성
### 약간의 시간이 필요.
wc = WordCloud(background_color='white',
               max_words=200,
               mask=Car_mask,
               contour_width=3,
               contour_color="steelblue",
               font_path=path).generate_from_frequencies(dict(data))
```

In [47]:

```
plt.figure(figsize=(12,8))
plt.imshow(wc)
plt.axis("off")
plt.show()
```

Ref

- https://bab2min.tistory.com/544 (https://bab2min.tistory.com/544)
- 한국어 불용어 참조 : https://www.ranks.nl/stopwords/korean (https://www.ranks.nl/stopwords/korean)
- https://www.lucypark.kr/courses/2015-dm/text-mining.html (https://www.lucypark.kr/courses/2015-dm/text-mining.html)