```
from bs4 import BeautifulSoup
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fonts-nanum is already the newest version (20200506-1).
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 12 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/usr/share/fonts/truetype: skipping, looped directory detected
/usr/share/fonts/truetype/humor-sans: skipping, looped directory detected
/usr/share/fonts/truetype/liberation: skipping, looped directory detected
/usr/share/fonts/truetype/nanum: skipping, looped directory detected
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
fc-cache: succeeded
```

```
!pip install konlpy
from konlpy.tag import Kkma
tokenizer=Kkma()
```

```
Requirement already satisfied: konlpy in /usr/local/lib/python3.10/dist-packages (0.6.0)
Requirement already satisfied: JPype1>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from konlpy) (1.4.1)
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from konlpy) (4.9.3)
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.10/dist-packages (from konlpy) (1.23.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from JPype1>=0.7.0->konlpy) (23.2)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
file_path = '/content/drive/MyDrive/Colab Notebooks/Textmining/대통령_취임사.csv'
```

```
import pandas as pd

# CSV 파일 읽기
df = pd.read_csv(file_path)

# 특정 열 두 개 선택
selected_df = df[['대통령', '연설내용']] # 실제 열 이름으로 수정 필요
```

```
from konlpy.tag import Kkma
kkma = Kkma()


# NaN 값을 제거
selected_df = selected_df.dropna(subset=['연설내용'])

# '연설내용' 열을 문자열로 변환
selected_df['연설내용'] = selected_df['연설내용'].astype(str)

# 특수 문자 제거 (한글과 공백만 남기기)
selected_df['연설내용'] = selected_df['연설내용'].apply(lambda x: re.sub('[^가-힣Ws]', '', x))


# '대통령' 열을 기준으로 데이터 그룹화
grouped = selected_df.groupby('대통령')
```

```
# 각 대통령별 연설문 토크나이징
tokenized_data = {}
for president, group in grouped:
    tokenized_texts = group['연설내용'].apply(lambda x: kkma.morphs(x))
    tokenized_data[president] = tokenized_texts.tolist()

# 결과 출력
for president, tokens in tokenized_data.items():
    print(f'대통령: {president}')
    for token_list in tokens:
        print(token_list)
```

```
대통령: 김대중
['존경', '하', '고', '사랑', '하', '는', '국민', '여러분', '오늘', '저', '는', '대한민국', '제대', '대통령', '에', '취임', '하', '게', '되', '었
대통령: 김영삼
['친애', '하', '는', '천', '만', '국내외', '동포', '여러분', '노태', '우', '대통령', '을', '비롯', '하', 'ㄴ', '전직', '대통령', '그리고', '이',
대통령: 노무현
['주한', '외교사', '절', '여러분', '그리', '고', '멀리서', '오신', '외빈', '여러', '분', '전두환', '전', '대통령', '과', '부', '요인', '을', '비
['존경', '하', '는', '국민', '여러분', '오늘', '저', '는', '대한민국', '의', '제대', '대통령', '에', '취임', '하', '기', '위하', '어', '이', '자
대통령: 노태우
['친애', '하', '는', '천', '만', '국내외', '동포', '여러분', '우리', '헌정', '발전', '을', '뒷받침', '하', '어', '주신', '윤보선', '최', '규', '
대통령: 문재인
['존경', '하', '고', '사랑', '하', '는', '국민', '여러분', '감사', '하', 'ㅂ니다', '국', '민', '여러분', '의', '위대', '하', 'ㄴ', '선택', '에',
['존경', '하', '는', '국민', '여러분', '임기', '년', '이', '남', '았', '습니다', '보통', '때', '이', '라면', '마무리', '를', '생각', '하', 'ㄹ',
대통령: 박근혜
['희망', '의', '새', '시대', '를', '열', '겠', '습니다', '존경', '하', '는', '국민', '여러분', '말', 'ㄴ', '해외', '동포', '여러분', '저', '는',
대통령: 박정희
['단군', '성조', '가', '천혜', '의', '이', '강토', '위', '에', '국기', '를', '닦', '으시', 'ㄴ지', '반만년', '연면', '히', '잇', '어', '오', 'ㄴ
['단군', '성조', '가', '천혜', '의', '이', '강토', '에', '국기', '를', '닦', '으시', 'ㄴ지', '반만년', '연면', '히', '잇', '어', '오', 'ㄴ', '역
['사랑', '하', '는', '천', '만', '국내외', '동포', '여러분', '그리', '고', '내외', '귀빈', '여러분', '저', '의', '차', '세계', '대전', '의', '포
['친애', '하', '는', '천만', '동포', '여러분', '그리', '고', '내외', '귀빈', '여러분', '우리', '는', '오늘', '고난', '과', '시련', '의', '역사',
['본인', '을', '제대', '대통령', '으로', '선출', '하', '어', '주신', '통일', '주체', '국민', '회의', '대의원', '과', '국민', '여러분', '에게', '
['친애', '하', '는', '천만', '동포', '여러분', '그리', '고', '내외', '귀빈', '여러', '분', '대망', '의', '년대', '를', '눈앞', '에', '바라보', '
대통령: 이명박
['존경', '하', '는', '국민', '여러분', '말', 'ㄴ', '해외', '동포', '여러분', '이', '자리', '에', '참석', '하', '시', 'ㄴ', '노무현', '김대중', '
대통령: 이승만
['여러', '번', '죽', '었', '더', 'ㄴ', '이', '몸', '이', '하나님', '의', '은혜', '와', '동포', '의', '애호', '로', '지금', '까지', '살아오', '다
['오늘', '취임식', '에서', '나', '가', '다시', '지게', '되', '는', '책임', '은', '내가', '하', 'ㄹ', '수', '만', '있', '으면', '지', '지', '않',
['나의', '사랑', '하', '는', '동포', '여러분', '내가', '오늘', '또', '한번', '우리', '민중', '앞', '에', '서서', '대통령', '취임', '선언식', '을
대통령: 전두환
['친애', '하', '는', '국민', '여러분', '내외', '귀빈', '여러분', '오늘', '새', '역사', '의', '장', '을', '열', '는', '뜻', '깊', '은', '식', '전
['친애', '하', '는', '국내외', '동포', '여러분', '그리', '고', '이', '자리', '를', '빛', '내', '주신', '내외', '귀빈', '여러분', '우리', '는', '
대통령: 최규하
['친애', '하', '는', '국민', '여러분', '이', '자리', '에', '참석', '하', '시', 'ㄴ', '내외', '귀빈', '여러분', '오늘', '본인', '은', '대한민국',
```

```
# 파일에서 불용어 불러오기
file_path = '/content/drive/MyDrive/Colab Notebooks/Textmining/mystopwords.txt'
with open(file_path, 'r', encoding='utf-8') as f:
    file_stopwords = [line.strip() for line in f.readlines()]

# 기본 불용어
basic_stopwords = [
    '하', '의', '을', '에', '는', '이', '를', '과', '도', '와', '으로', '에서', '위하','대하','시다', '로', '에게', '라', '만', '게','고자', '로서', '
]

# 두 불용어 리스트 합치기
stopwords = file_stopwords + basic_stopwords

# 중복된 불용어 제거
stopwords = list(set(stopwords))
```

```
from collections import Counter

pos_frequencies = {}

# 대통령별로 토큰과 해당 토큰의 품사를 묶어서 빈도 계산
for president, token_lists in tokenized_data.items():
    token_pos_counts = Counter()

    for tokens in token_lists:
        pos_tagged_tokens = kkma.pos(" ".join(tokens))

        # 길이가 두 글자 이상이며 불용어에 속하지 않는 토큰만 포함
        filtered_tokens = [
            (token, pos) for token, pos in pos_tagged_tokens
            if len(token) >= 2 and token not in stopwords
        ]

        token_pos_counts.update(filtered_tokens)

    sorted_pos_counts = sorted(token_pos_counts.items(), key=lambda x: x[1], reverse=True)
```

```
    pos_frequencies[president] = sorted_pos_counts


# 결과 출력
for president, sorted_pos_counts in pos_frequencies.items():
    for (token, pos), freq in sorted_pos_counts:
        print(f'대통령: {president}, 단어: {token}, 품사: {pos}, 빈도: {freq}')
```

```
대통령: 최규하, 단어: 권리, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 의무, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 서로, 품사: MAG, 빈도: 1
대통령: 최규하, 단어: 균형, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 이루, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 문명, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 불가, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 요건, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 법치, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 자각, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 긍지, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 이렇, 품사: VA, 빈도: 1
대통령: 최규하, 단어: 자선, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 기약, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 순조, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 참여, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 영역, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 가지, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 모든, 품사: MDT, 빈도: 1
대통령: 최규하, 단어: 지혜, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 동원, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 대로, 품사: NNB, 빈도: 1
대통령: 최규하, 단어: 과의, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 기본, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 자문, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 정계, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 원로, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 중진, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 그리, 품사: MAG, 빈도: 1
대통령: 최규하, 단어: 인격, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 경비, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 구성, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 기구, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 만들, 품사: VV, 빈도: 1
대통령: 최규하, 단어: 용의, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 민족, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 장구, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 무수, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 역경, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 극복, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 스스로, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 생존, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 전통, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 이제, 품사: MAG, 빈도: 1
대통령: 최규하, 단어: 한번, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 시련기, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 직면, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 대동, 품사: NNP, 빈도: 1
대통령: 최규하, 단어: 단결, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 건설, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 준비, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 갖추, 품사: MAG, 빈도: 1
대통령: 최규하, 단어: 방울, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 모이, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 도도, 품사: MAG, 빈도: 1
대통령: 최규하, 단어: 영광, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 창조, 품사: NNG, 빈도: 1
대통령: 최규하, 단어: 전진, 품사: NNG, 빈도: 1
```

```
# 결과 출력
for president, sorted_pos_counts in pos_frequencies.items():
    print(f'대통령: {president}')
    for (token, pos), freq in sorted_pos_counts[:20]:  # 상위 20개만 출력
        print(f'단어: {token}, 품사: {pos}, 빈도: {freq}')
    print("-" * 50)  # 구분선
```

```
대통령: 문재인
단어: 국민, 품사: NNG, 빈도: 47
단어: 본인, 품사: NNG, 빈도: 42
단어: 어야, 품사: ECD, 빈도: 42
단어: 읍니다, 품사: EFN, 빈도: 37
단어: 국가, 품사: NNG, 빈도: 28
단어: 어서, 품사: ECD, 빈도: 27
단어: 사회, 품사: NNG, 빈도: 24
단어: 정부, 품사: NNG, 빈도: 24
단어: 시대, 품사: NNG, 빈도: 20
단어: 민족, 품사: NNG, 빈도: 20
단어: 발전, 품사: NNG, 빈도: 16
단어: 복지, 품사: NNG, 빈도: 15
단어: 민주, 품사: NNG, 빈도: 15
단어: 역사, 품사: NNG, 빈도: 14
단어: 정치, 품사: NNG, 빈도: 14
단어: 대통령, 품사: NNG, 빈도: 13
단어: 나라, 품사: NNG, 빈도: 12
단어: ㅂ니다, 품사: EFN, 빈도: 12
단어: 문화, 품사: NNG, 빈도: 12
단어: 해방, 품사: NNG, 빈도: 12
--------------------------------------------------
대통령: 최규하
단어: 국민, 품사: NNG, 빈도: 25
단어: 헌법, 품사: NNG, 빈도: 21
단어: 본인, 품사: NNG, 빈도: 17
단어: 국가, 품사: NNG, 빈도: 17
단어: 경제, 품사: NNG, 빈도: 16
단어: 안정, 품사: NNG, 빈도: 13
단어: 어서, 품사: ECD, 빈도: 13
단어: 발전, 품사: NNG, 빈도: 12
단어: 문제, 품사: NNG, 빈도: 11
단어: 사회, 품사: NNG, 빈도: 11
단어: 어야, 품사: ECD, 빈도: 11
단어: 정부, 품사: NNG, 빈도: 10
단어: 으며, 품사: ECE, 빈도: 8
단어: 정치적, 품사: NNG, 빈도: 8
단어: 개정, 품사: NNG, 빈도: 8
단어: 대통령, 품사: NNG, 빈도: 7
단어: 국제, 품사: NNG, 빈도: 7
단어: ㅂ니다, 품사: EFN, 빈도: 7
단어: 협력, 품사: NNG, 빈도: 7
단어: 관계, 품사: NNG, 빈도: 7
--------------------------------------------------
```

```
top_nouns_by_president = {}

# 각 대통령별로 'N'으로 시작하는 품사의 토큰만 상위 20개 선택
for president, sorted_pos_counts in pos_frequencies.items():
    top_nouns = [(token, freq) for (token, pos), freq in sorted_pos_counts if pos.startswith('N')]
    top_nouns_by_president[president] = top_nouns

# 결과 출력
for president, top_nouns in top_nouns_by_president.items():
    print(f'대통령: {president}')
    for token, freq in top_nouns:
        print(f'단어: {token}, 빈도: {freq}')
    print('-' * 50)  # 구분선
```

```
단어: 권노, 빈도: 1
단어: 중진, 빈도: 1
단어: 인격, 빈도: 1
단어: 경비, 빈도: 1
단어: 구성, 빈도: 1
단어: 기구, 빈도: 1
단어: 용의, 빈도: 1
단어: 민족, 빈도: 1
단어: 장구, 빈도: 1
단어: 무수, 빈도: 1
단어: 역경, 빈도: 1
단어: 극복, 빈도: 1
단어: 스스로, 빈도: 1
단어: 생존, 빈도: 1
단어: 전통, 빈도: 1
단어: 한번, 빈도: 1
단어: 시련기, 빈도: 1
단어: 직면, 빈도: 1
단어: 대동, 빈도: 1
단어: 단결, 빈도: 1
단어: 건설, 빈도: 1
단어: 준비, 빈도: 1
단어: 방울, 빈도: 1
단어: 모이, 빈도: 1
단어: 영광, 빈도: 1
단어: 창조, 빈도: 1
단어: 전진, 빈도: 1
--------------------------------------------------
```

```
!pip install wordcloud

from wordcloud import WordCloud
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm


font_path = '/usr/share/fonts/truetype/nanum/NanumGothic.ttf'
font_name = fm.FontProperties(fname=font_path, size=10).get_name()
plt.rc('font', family=font_name)

for president, top_nouns in top_nouns_by_president.items():
    # 워드클라우드 설정
    wc = WordCloud(
        font_path=font_path,  # 폰트 경로
        background_color='white',  # 배경색 설정
        width=300,
        height=300
    )

    # 워드클라우드 생성
    wordcloud = wc.generate_from_frequencies(dict(top_nouns))

    # 그래프 설정 및 표시
    plt.figure(figsize=(5, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(f'대통령: {president}')
    plt.axis('off')
    plt.show()
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.10/dist-packages (1.9.2)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.10/dist-packages (from wordcloud) (1.23.5)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from wordcloud) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

대통령: 김대중

대통령: 김영삼

대통령: 노무현

대통령: 노태우

대통령: 문재인

대통령: 박근혜

대통령: 박정희

대통령: 이명박

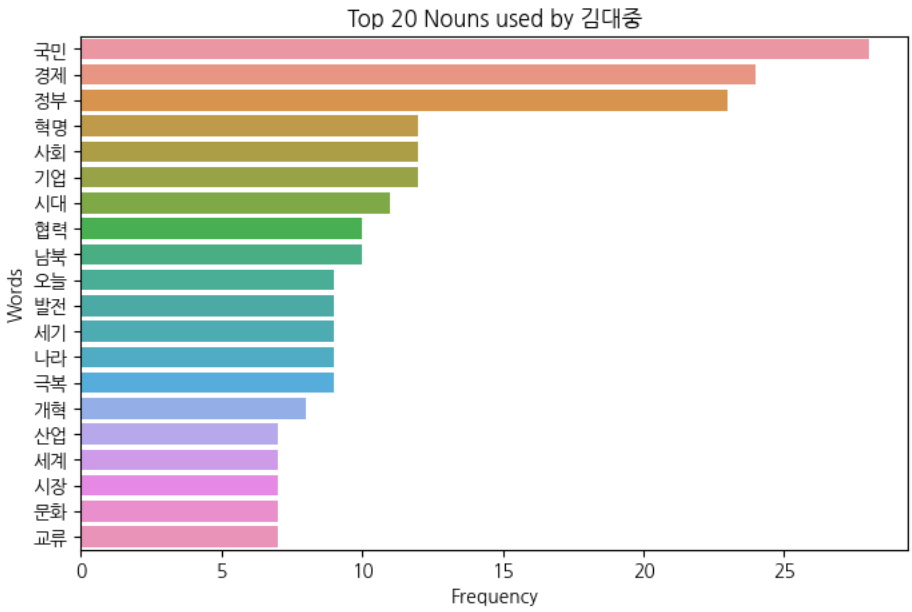
대통령: 이승만

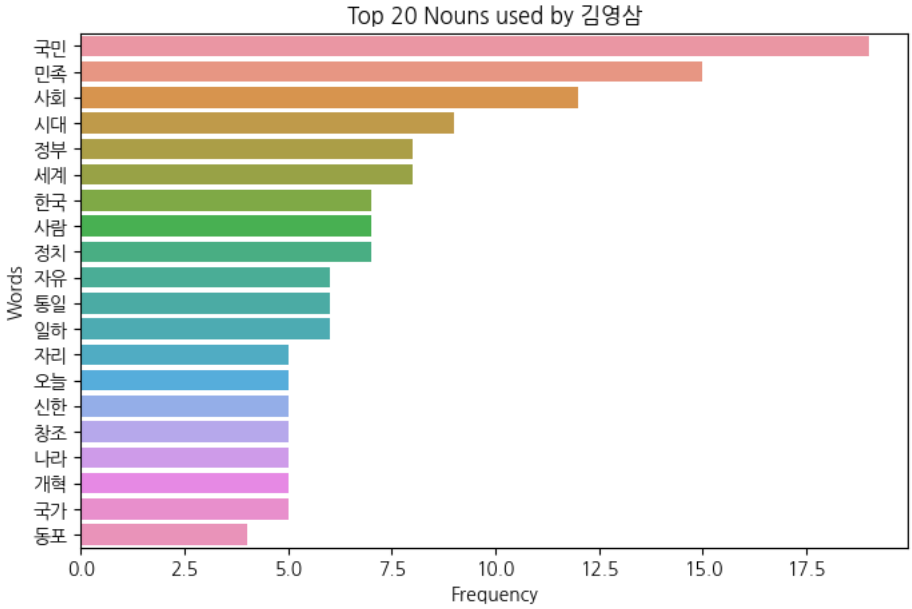
대통령: 전두환

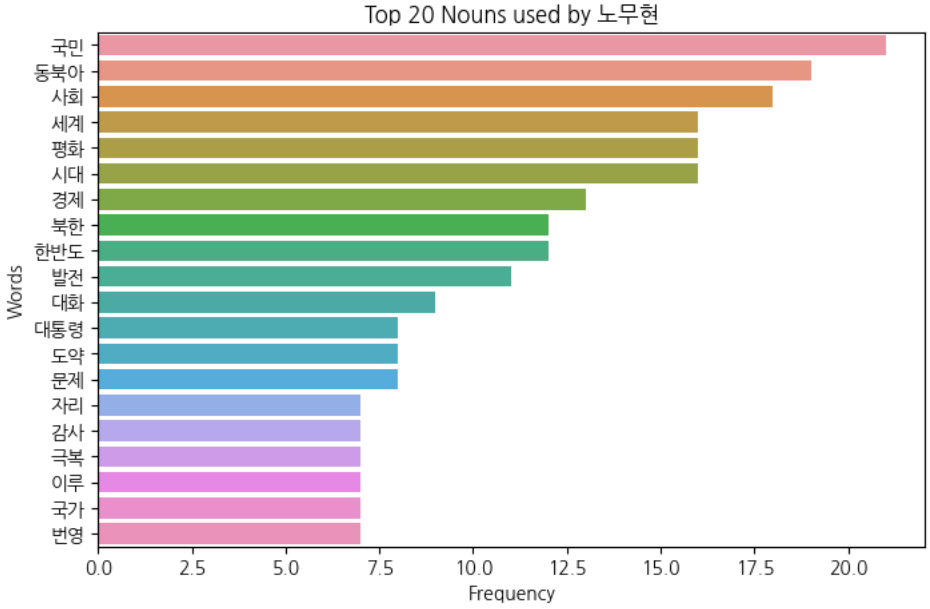
```
top_20_nouns_by_president = {}

# 각 대통령별로 'N'으로 시작하는 품사의 토큰만 상위 20개 선택
for president, sorted_pos_counts in pos_frequencies.items():
    top_20_nouns = [(token, freq) for (token, pos), freq in sorted_pos_counts if pos.startswith('N')][:20]
    top_20_nouns_by_president[president] = top_20_nouns

# 결과 출력
for president, top_nouns in top_20_nouns_by_president.items():
    print(f'대통령: {president}')
    for token, freq in top_nouns:
        print(f'단어: {token}, 빈도: {freq}')
    print('-' * 50)  # 구분선
```

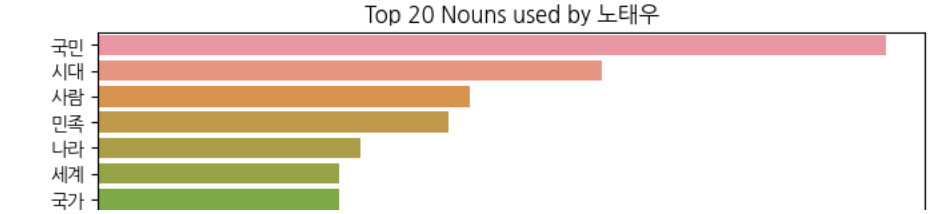
```
단어: 모든, 빈도: 12
단어: 정부, 빈도: 12
단어: 동포, 빈도: 10
단어: 지금, 빈도: 10
단어: 생각, 빈도: 10
단어: 국민, 빈도: 10
단어: 주의, 빈도: 10
단어: 통일, 빈도: 10
단어: 목적, 빈도: 10
단어: 나의, 빈도: 9
단어: 직책, 빈도: 9
단어: 국회, 빈도: 9
단어: 공산당, 빈도: 9
--------------------------------------------------
대통령: 전두환
단어: 국민, 빈도: 47
단어: 본인, 빈도: 42
단어: 국가, 빈도: 28
단어: 사회, 빈도: 24
단어: 정부, 빈도: 24
단어: 시대, 빈도: 20
단어: 민족, 빈도: 20
단어: 발전, 빈도: 16
단어: 복지, 빈도: 15
단어: 민주, 빈도: 15
단어: 역사, 빈도: 14
단어: 정치, 빈도: 14
단어: 대통령, 빈도: 13
단어: 나라, 빈도: 12
단어: 문화, 빈도: 12
단어: 해방, 빈도: 12
단어: 생활, 빈도: 12
단어: 오늘, 빈도: 11
단어: 경제, 빈도: 11
단어: 북한, 빈도: 11
--------------------------------------------------
대통령: 최규하
단어: 국민, 빈도: 25
단어: 헌법, 빈도: 21
단어: 본인, 빈도: 17
단어: 국가, 빈도: 17
단어: 경제, 빈도: 16
단어: 안정, 빈도: 13
단어: 발전, 빈도: 12
단어: 문제, 빈도: 11
단어: 사회, 빈도: 11
단어: 정부, 빈도: 10
단어: 정치적, 빈도: 8
단어: 개정, 빈도: 8
단어: 대통령, 빈도: 7
단어: 국제, 빈도: 7
단어: 협력, 빈도: 7
단어: 관계, 빈도: 7
단어: 노력, 빈도: 7
단어: 질서, 빈도: 6
단어: 나라, 빈도: 6
단어: 추진, 빈도: 6
--------------------------------------------------
```

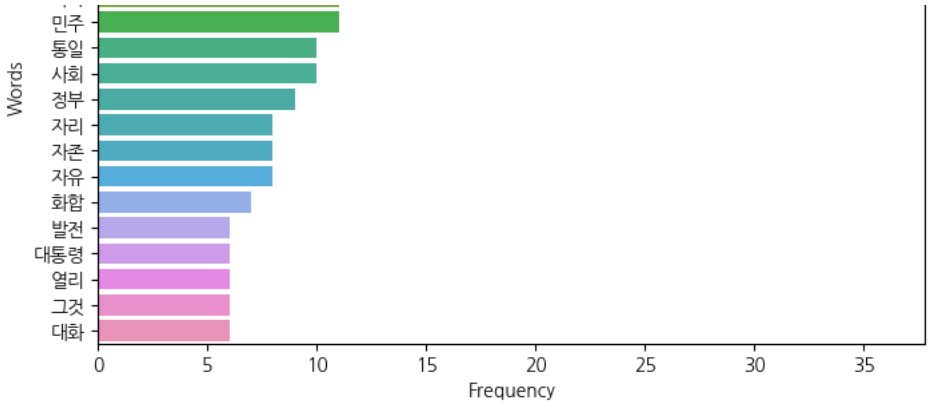
```python
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

import seaborn as sns

path = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'
fontprop = fm.FontProperties(fname=path, size=5)
plt.rc('font', family='NanumGothic')


for president, top_nouns in top_20_nouns_by_president.items():
    plt.figure(figsize=(8, 5))  # 그래프 크기 설정

    # 데이터 분리
    words, frequencies = zip(*top_nouns)

    # 그래프 그리기
    sns.barplot(x=list(frequencies), y=list(words))
    plt.title(f"Top 20 Nouns used by {president}")  # 제목 설정
    plt.xlabel('Frequency')  # x축 라벨 설정
    plt.ylabel('Words')      # y축 라벨 설정
    plt.show()
```

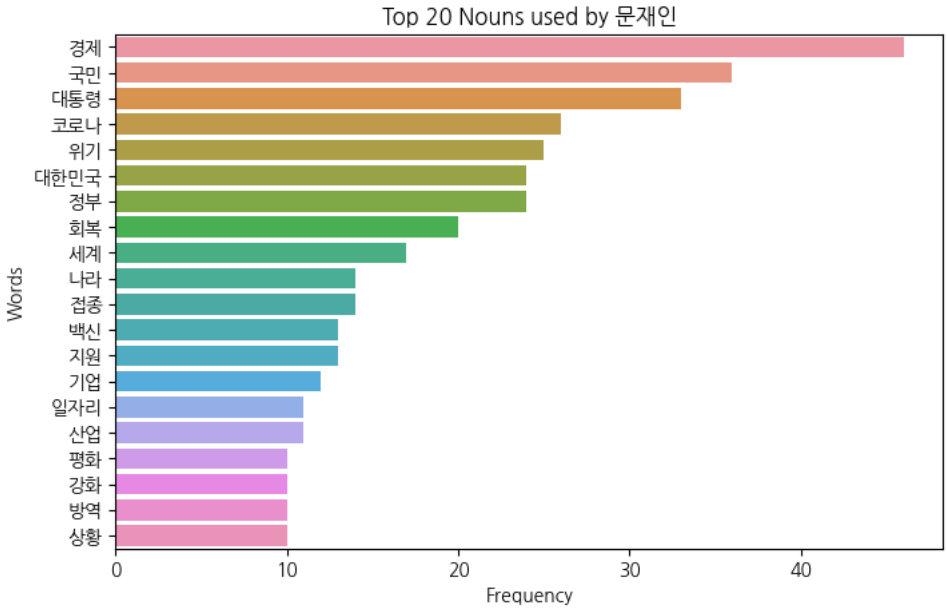

Top 20 Nouns used by 김대중
국민
경제
정부
혁명
사회
기업
시대
협력
남북
오늘
발전
세기
나라
극복
개혁
산업
세계
시장
문화
교류
Words
0
5
10
15
20
25
Frequency

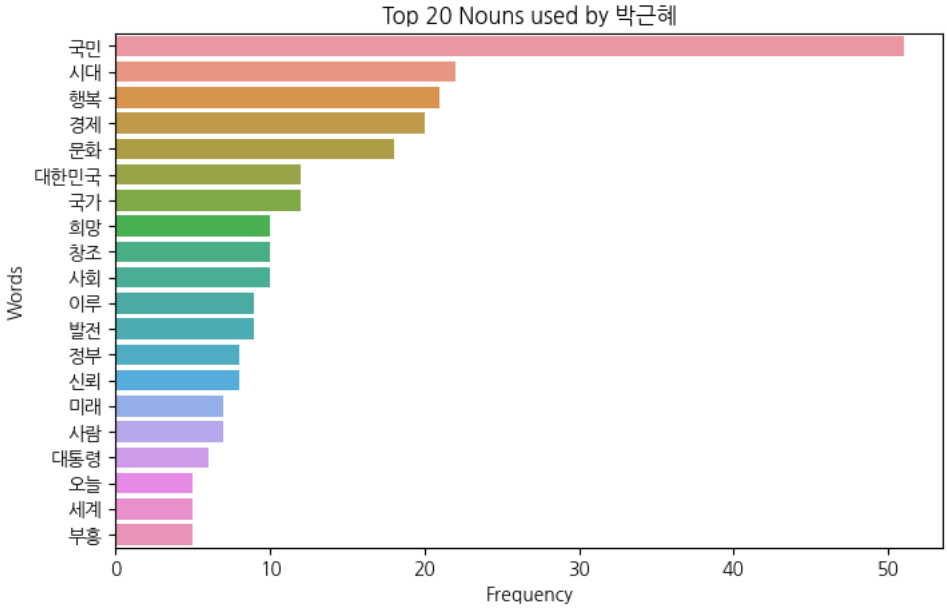

Top 20 Nouns used by 김영삼
국민
민족
사회
시대
정부
세계
한국
사람
정치
자유
통일
일하
자리
오늘
신한
창조
나라
개혁
국가
동포
Words
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Frequency

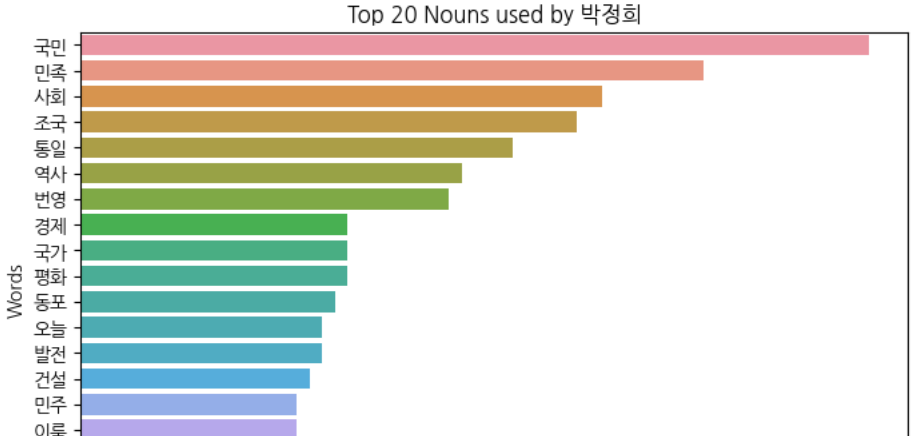

Top 20 Nouns used by 노무현
국민
동북아
사회
세계
평화
시대
경제
북한
한반도
발전
대화
대통령
도약
문제
자리
감사
극복
이루
국가
번영
Words
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Frequency

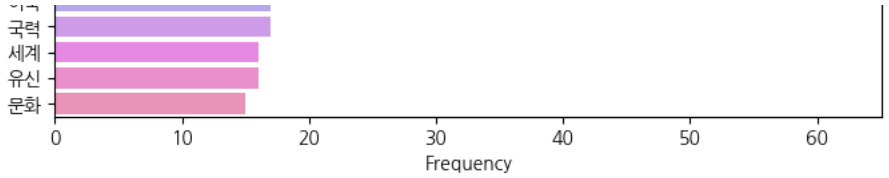

Top 20 Nouns used by 노태우
국민
시대
사람
민족
나라
세계
국가

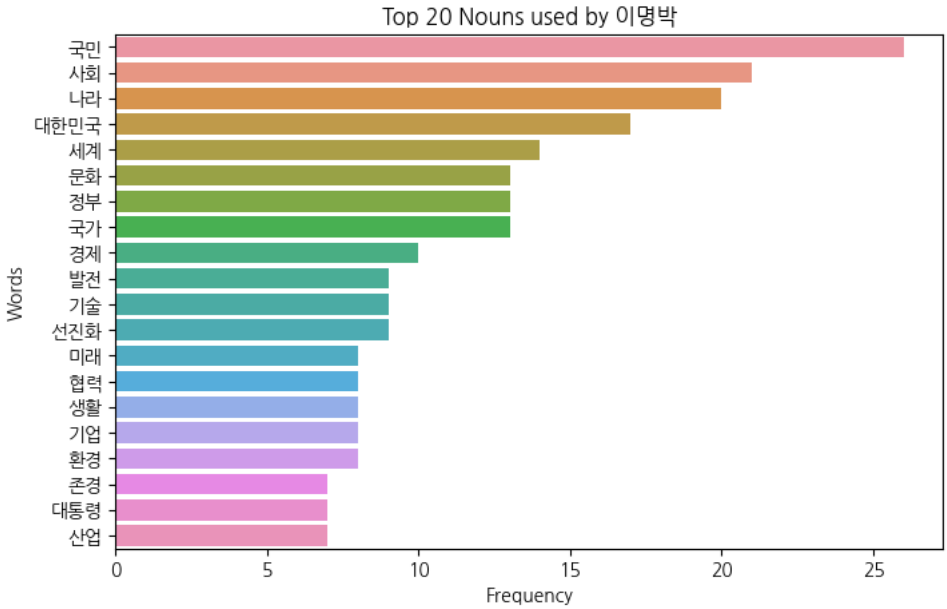

민주
통일
사회
정부
자리
자존
자유
화합
발전
대통령
열리
그것
대화
Words
0
5
10
15
20
25
30
35
Frequency

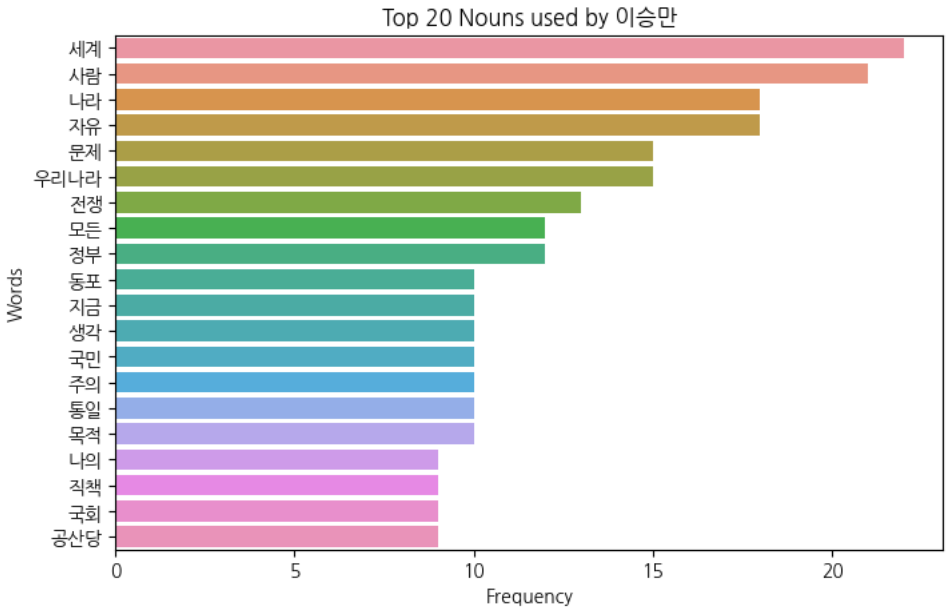

Top 20 Nouns used by 문재인
경제
국민
대통령
코로나
위기
대한민국
정부
회복
세계
나라
접종
백신
지원
기업
일자리
산업
평화
강화
방역
상황
Words
0
10
20
30
40
Frequency

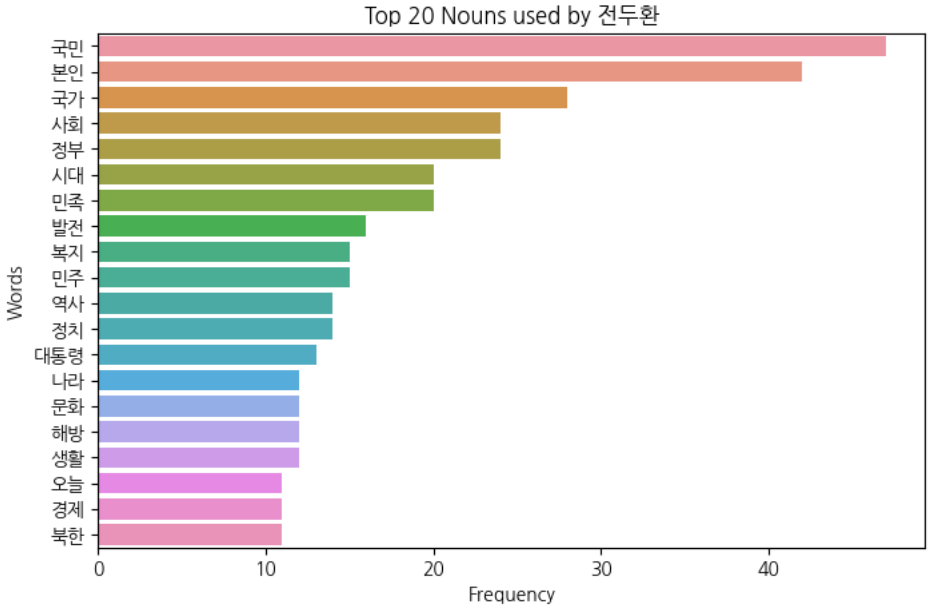

Top 20 Nouns used by 박근혜
국민
시대
행복
경제
문화
대한민국
국가
희망
창조
사회
이루
발전
정부
신뢰
미래
사람
대통령
오늘
세계
부흥
Words
0
10
20
30
40
50
Frequency


Top 20 Nouns used by 박정희
국민
민족
사회
조국
통일
역사
번영
경제
국가
평화
동포
오늘
발전
건설
민주
이룩
Words


국력
세계
유신
문화
0
10
20
30
40
50
60
Frequency


Top 20 Nouns used by 이명박
Words
국민
사회
나라
대한민국
세계
문화
정부
국가
경제
발전
기술
선진화
미래
협력
생활
기업
환경
존경
대통령
산업
0
5
10
15
20
25
Frequency


Top 20 Nouns used by 이승만
Words
세계
사람
나라
자유
문제
우리나라
전쟁
모든
정부
동포
지금
생각
국민
주의
통일
목적
나의
직책
국회
공산당
0
5
10
15
20
Frequency


Top 20 Nouns used by 전두환
Words
국민
본인
국가
사회
정부
시대
민족
발전
복지
민주
역사
정치
대통령
나라
문화
해방
생활
오늘
경제
북한
0
10
20
30
40
Frequency


Top 20 Nouns used by 최규하