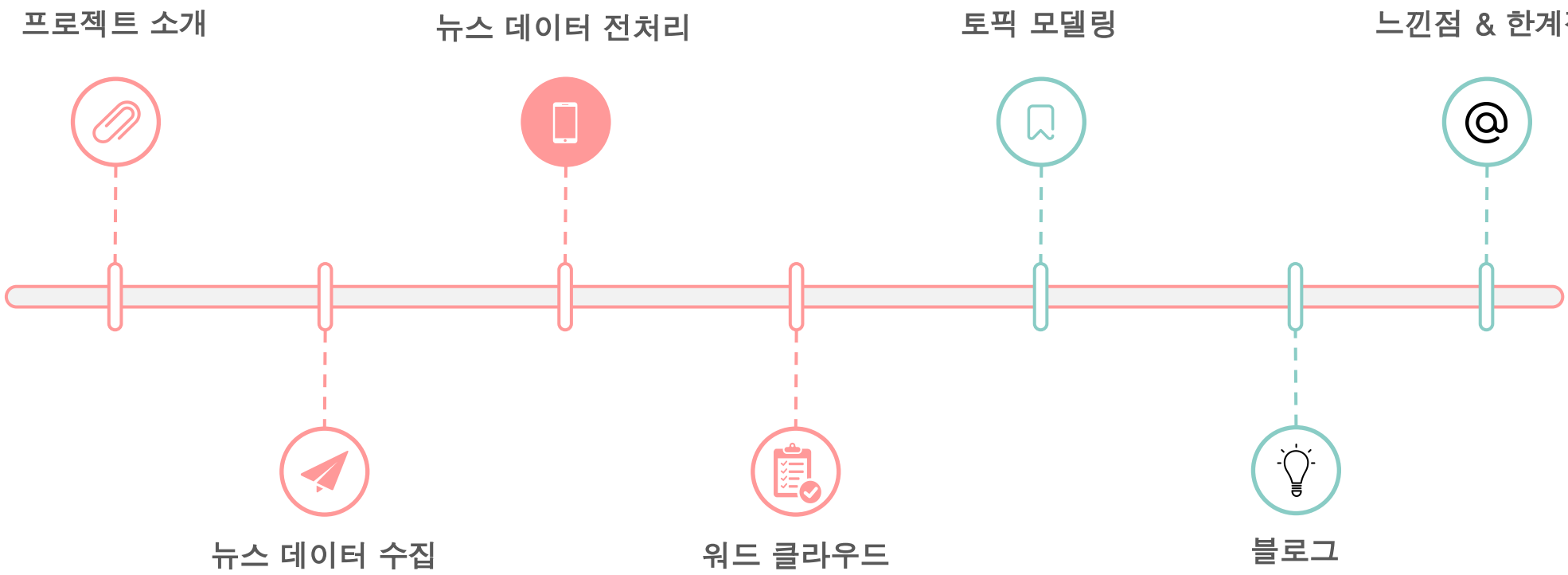
코로나 백신 관련 뉴스 기사 및 블로그 토픽 모델링

정현석 정소비 정연규

목차



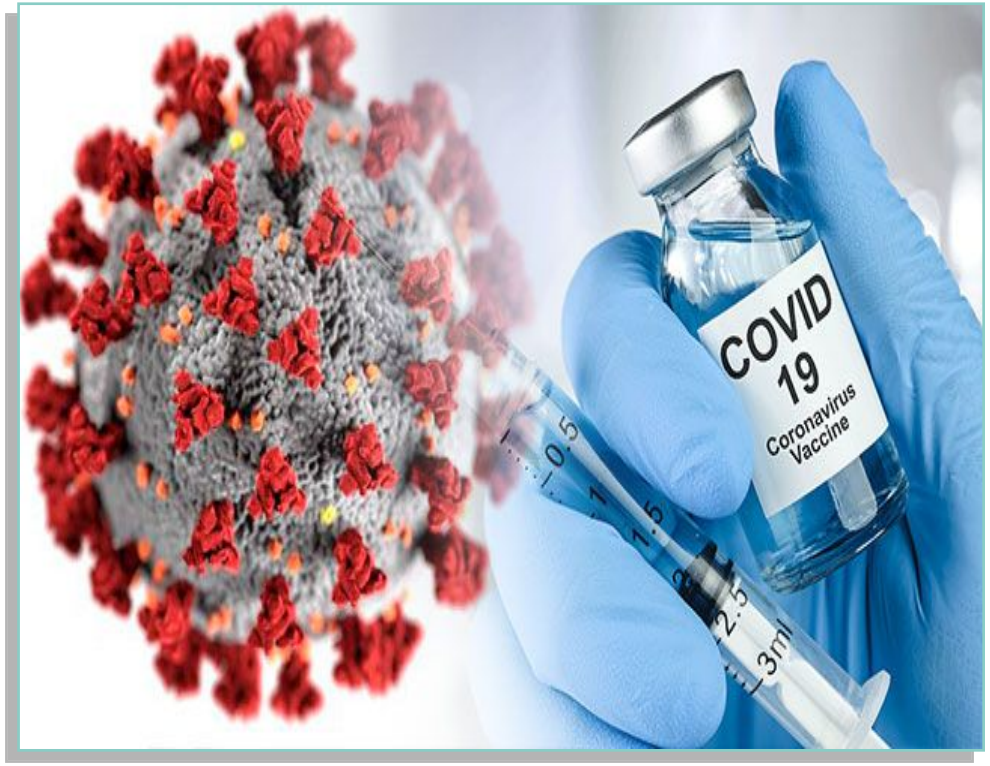
주제 선정 배경 및 프로젝트 소개

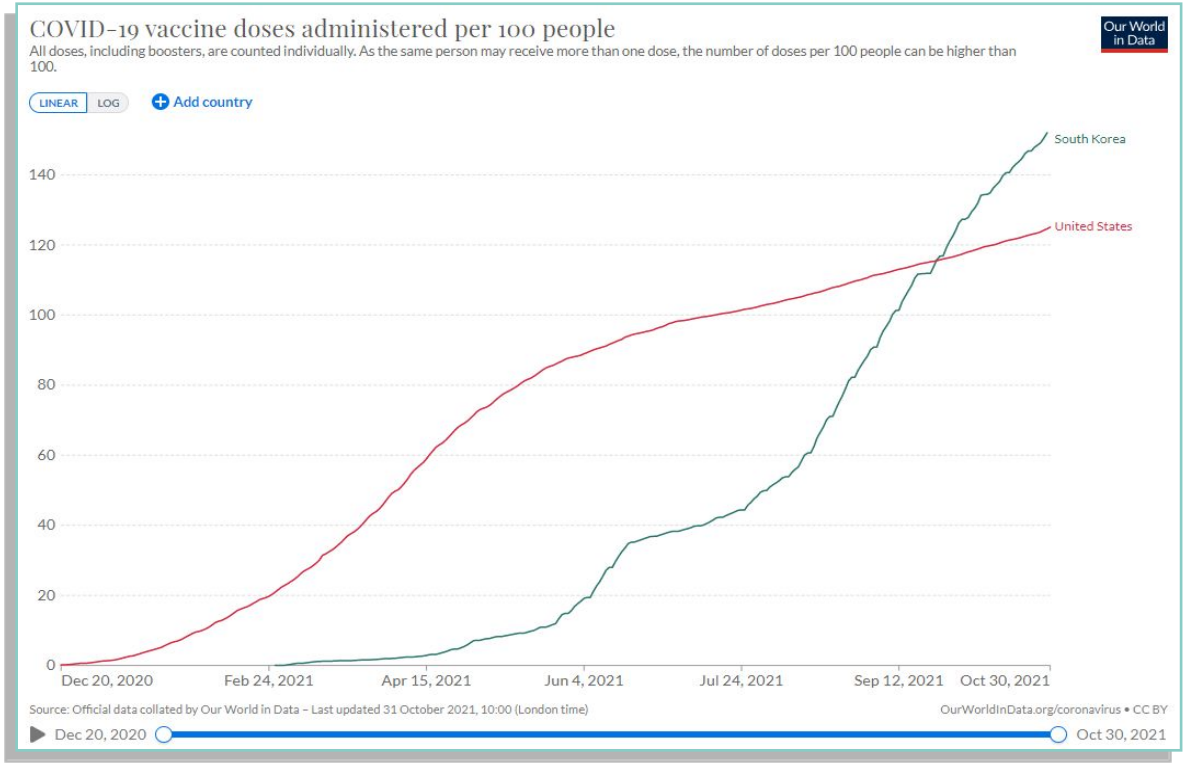
주제 선정 배경

프로젝트 소개

프로젝트 소개

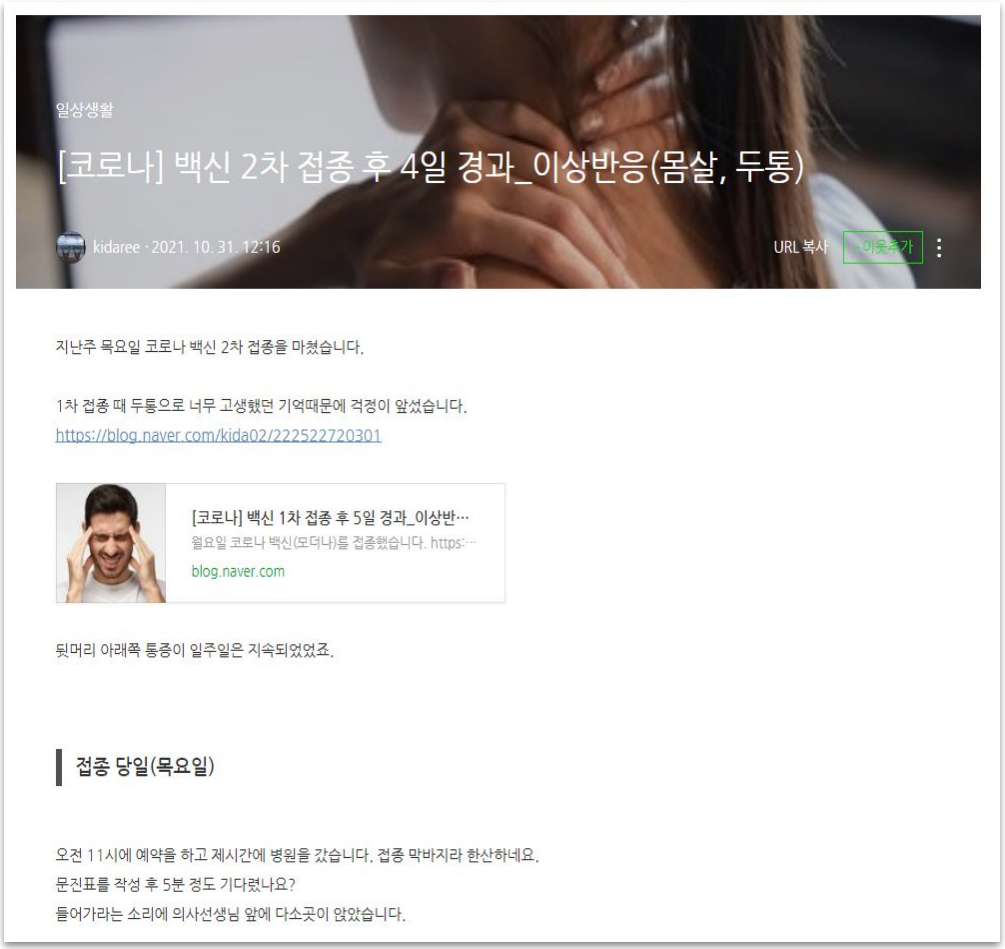
주제 선정 배경





코로나 백신이 도입됨에 따라 많은 사람들이 접종을 진행중
하지만 코로나 백신 접종에 대해 불안감을 호소하는 사람들이 존재

프로젝트 소개



코로나 백신에 대해 국내 언론이 보도하는 내용과 실제 여론 간의 차이,
한국과 미국 언론간의 보도 내용 차이 등을 조사하고 분석

뉴스 데이터 수집

한국 뉴스 데이터 수집

미국 뉴스 데이터 수집

뉴스 데이터 수집 - 한국 네이버 뉴스

API 구현 예제

```
#-*- coding: utf-8 -*-
import os
import sys
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
url = "https://openapi.naver.com/v1/datalab/search";
body = "{\"startDate\":\"2017-01-01\",\"endDate\":\"2017-04-30\",\"timeUnit\":\"month\",\"keywordGroups\":[{\"groupName\":\"한글

request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
request.add_header("Content-Type","application/json")
response = urllib.request.urlopen(request, data=body.encode("utf-8"))
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

요청 변수

요청 변수	타입	필수 여부	기본값	설명
query	string	Y	-	검색을 원하는 문자열로서 UTF-8로 인코딩한다.
display	integer	N	10(기본값), 100(최대)	검색 결과 출력 건수 지정
start	integer	N	1(기본값), 1000(최대)	검색 시작 위치로 최대 1000까지 가능
sort	string	N	sim, date(기본값)	정렬 옵션: sim (유사도순), date (날짜순)

필드	타입	설명
item/items	-	XML 포멧에서는 item 태그로, JSON 포멧에서는 items 속성으로 표현된다. 개별 검색 결과이며 title, originallink, link, description, pubDate를 포함한다.
title	string	개별 검색 결과이며, title, originallink, link, description, pubDate 를 포함한다.
originallink	string	검색 결과 문서의 제공 언론사 하이퍼텍스트 link를 나타낸다.
link	string	검색 결과 문서의 제공 네이버 하이퍼텍스트 link를 나타낸다.
description	string	검색 결과 문서의 내용을 요약한 패시지 정보이다. 문서 전체의 내용은 link를 따라가면 읽을 수 있다. 패시지에서 검색어와 일치하는 부분은 **태그로 감싸져 있다.**
pubDate	datetime	검색 결과 문서가 네이버에 제공된 시간이다.

네이버 **api**를 통해 뉴스 크롤링 진행 (예시)

뉴스 데이터 수집 - 한국 네이버 뉴스

```python
import os
import sys
import urllib.request
import requests

news_data = []
page_count = 100

client_id =
client_secret =
encText = urllib.parse.quote("코로나 백신")

for idx in range(page_count):
    # json 결과
    url = "https://openapi.naver.com/v1/search/news?query=" + encText + "&start=" + str(idx * 10 + 1)
    # url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
    request = urllib.request.Request(url)
    request.add_header("X-Naver-Client-Id",client_id)
    request.add_header("X-Naver-Client-Secret",client_secret)
    response = urllib.request.urlopen(request)
    rescode = response.getcode()

    if(rescode==200):
    #    response_body = response.read()
        result = requests.get(response.geturl(),
                              headers={"X-Naver-Client-Id":client_id,
                                       "X-Naver-Client-Secret":client_secret}
                             )
        news_data.append(result.json())
    #    print(response_body.decode('utf-8'))
    else:
        print("Error Code:" + rescode)
```

```python
print(naver_news_title[0])
print("================================")
print(naver_news_content[0])
```

```
  0%|          | 0/6 [00:00<?, ?it/s]

https://sports.news.naver.com/news.nhn?oid=469&aid=0000637247
https://sports.news.naver.com/news.nhn?oid=119&aid=0002541912
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=100&oid=003&aid=0010793400
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=100&oid=123&aid=0002257899
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=102&oid=003&aid=0010793399
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=104&oid=030&aid=0002977725

  0%|          | 0/6 [00:00<?, ?it/s]

https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=101&oid=008&aid=0004661815
https://sports.news.naver.com/news.nhn?oid=382&aid=0000940711
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=104&oid=025&aid=0003145590
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=102&oid=468&aid=0000796444
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=102&oid=003&aid=0010793385
https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=105&oid=023&aid=0003648793

  0%|          | 0/6 [00:00<?, ?it/s]

https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=102&oid=421&aid=0005682128
```

네이버 **api**를 통해 뉴스 크롤링 진행 (실전)

뉴스 데이터 수집 - 한국 네이버 뉴스

Crawling Code

```
    ###### 뉴스 타이틀 긁어오기 ######

    title = None

    try:
        item = soup.find('div', class_="article_info")
        title = item.find('h3', class_="tts_head").get_text()
        #print(title)

    except:
        title = "OUTLINK"

    #print(title)
    news_page_title.append(title)


    ###### 뉴스 본문 긁어오기 ######

    doc = None
    text = ""

    data = soup.find_all("div", {"class" : "_article_body_contents"})
    if data:
        for item in data:

            text = text + str(item.find_all(text=True)).strip()
            text = ast.literal_eval(text)
            doc = ' '.join(text)

    else:
        doc = "OUTLINK"

    news_page_content.append(doc.replace('₩n', ' '))


naver_news_title.append(news_page_title)
naver_news_content.append(news_page_content)

time.sleep(2)
```

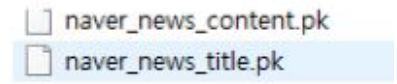
출력 결과

```
print(naver_news_content[0])
```

```
['OUTLINK', 'OUTLINK', '   본문 내용      TV플레이어      // TV플레이어      // flash 오
llback() {}    기사내용 요약 "하나 된 아세안, 팬데믹 대응 연대·협력 모범"  "韓, 아세
진아 기자 = 문재인 대통령이 26일 청와대 충무실에서 열린 제22차 한-아세안 화상 정상회
[서울=뉴시스]김태규 기자 = 문재인 대통령은 26일 "역내포괄적경제동반자협정(RCEP) 비준
협정(FTA)과 한·필리핀 FTA도 조속히 발효되도록 최선을 다하겠다"고 밝혔다. 문 대통령은
여한 제22차 한·아세안(ASEAN·동남아시아국가연합) 정상회의 모두 발언에서 "한·아세안
한 우리의 한층 강화된 의지를 천명하게 돼 매우 뜻깊다"며 이렇게 말했다. RCEP은 동남아
·호주·인도·뉴질랜드 6개국을 더해 아시아·태평양 지역 16개국 사이의 무역의 룰을 정
가 유명무실화 된 환태평양경제동반자협정(CPTPP)보다 참여국이 5개국이 많아 ₩'세계 최대
들은 2019년 11월 제3차 RCEP 정상회의에서 RCEP 협정문을 타결했다. 이후 각국들은 자국
지지 않고 있다. 문 대통령은 "우리는 코로나 극복과 더 나은 회복을 위해 더 깊은 우정을
심으로 델타변이가 퍼지고, 빈번한 생산 차질로 세계경제 회복이 제약받고 있다"고 진단했
은 나라의 어려움으로 연결되고, 결국 연대와 협력만이 코로나 극복의 길이라는 것을 증명
통령은 "아세안은 지난 반세기 하나의 공동체로 발전하며 위기를 기회로 바꿔왔다"며 "팬
현하고 연대와 협력의 모범이 됐다"고 평가했다. [서울=뉴시스] 김진아 기자 = 문재인 대
회의에서 기념촬영을 하고 있다. 2021.10.26. bluesoda@newsis.com   이어 "아세안과 한국
다. 한국은 아세안과의 관계를 매우 중시한다"며 "한국은 아세안의 친구로서 코로나를 함
나갈 것"이라고 밝혔다. 또 "한국은 2017년부터 이어온 신남방정책과 지난해 발표한 신남
다"며 "한국은 코로나 아세안 대응 기금에 500만 달러를 추가로 공여하여 아세안의 백신
화하여 아세안의 경제 회복에 기여할 것"이라고 강조했다. 앞서 아세안 의장국인 하싸날
의 가장 중요한 전략적인 동반자 중 하나"라며 "한·아세안 관계가 많은 진전을 이루고 있
적으로 500만 불의 추가 공여를 통해서 코로나19 아시아 대응 기금을 통해 공여해 주신 것
략적인 동반자 관계를 추가적으로 강화하고, 보다 회복력 있고 지속가능한 미래를 함께 건
본문 내용      TV플레이어      // TV플레이어      // flash 오류를 우회하기 위한 함수 추
7 일까지 이틀간 화상 정상회의로 진행 예정 "역내 백신보급 및 지원 강화할 것...대응 기
통령의 신남방정책...₩'한-아세안 실질 협력 추진₩' 높이 평가 文대통령, 한반도 완전한 비
문재인 대통령이  26 일 청와대 충무실에서 열린 한-아세안 화상 정상회의에서 기념촬영을
오후 화상으로 개최된 제 22 차 한-아세안 정상회의에 참석, 이틀간 예정된 아세안 관련 정
서 코로나를 함께 극복하고, 포용적이며 지속가능한 미래를 함께 만들어 나갈 것임을 강조
난 4년간 신남방정책 협력 성과를 종합 점검하고, 코로나 19  위기를 넘어 보다 나은 미래
논의했다.    문 대통령은 모두 발언에서 한국은 아세안의 신뢰할 수 있는 파트너로서 이
```

네이버 뉴스 데이터 저장

```
with open("naver_news_title.pk", "wb") as f:
    pickle.dump(naver_news_title, f)

with open("naver_news_content.pk", "wb") as f:
    pickle.dump(naver_news_content, f)
```



뉴스 타이틀과 본문을 각각 긁어옴 (출력 결과 확인 후 **pickle**로 데이터 저장)

뉴스 데이터 수집 - 미국 CNN 뉴스

미국 뉴스(CNN) 데이터 크롤링

```
title_list_list=[]
date_list=[]
body_list=[]

while(1):

    # 검색 날짜 지정
    if (min_day.month in [1,3,5,7,8,10,12]):
        delta = relativedelta(days=31)
    elif (min_day.month in [4,6,9,11]):
        delta = relativedelta(days=30)
    else:
        delta = relativedelta(days=28)

    max_day = max_day - delta
    min_day = max_day
    min_day = max_day.replace(day=1)

    if (min_day.month not in [2]):
        break
```

```
# 검색 url 지정
url_start = "https://www.google.com/search?q=covid+19+cnn+~vaccine+OR+~vaccination&biw=1920&bih=924&source=lnt&tbs=cdr%3A

# url의 년,월,일 지정
url_min_month = str(min_day.month)
url_min_day = str(min_day.day)
url_min_year = str(min_day.year)
url_max_start = "%2Ccd_max%3A"
url_max_month = str(max_day.month)
url_max_day = str(max_day.day)
url_max_year = str(max_day.year)
url_end = "&tbm=nws"
between = "%2F"

url = url_start + url_min_month + between + url_min_day + between + url_min_year + url_max_start + url_max_month + betwe



# 해당 날짜의 url에 접속
driver.get(url)
html=driver.page_source
soup=BeautifulSoup(html, 'html.parser')
is_cnn=soup.select("div.CEMjEf > span")#rso > div:nth-child(3) > g-card > div > div > a > div > div.iRPxbe > div.CEMjEf >
```

검색 url 지정 및 접속

뉴스 데이터 수집 - 미국 CNN 뉴스

미국 뉴스(CNN) 데이터 크롤링

```
# 뉴스 기사 50개 받아오기
while(len(title_list)<50):
    html=driver.page_source
    soup=BeautifulSoup(html, 'html.parser')

    i=i+1

    # 8개의 기사를 받아오면 다음 페이지로 넘어감
    if i%8==0:
        html=driver.page_source
        soup=BeautifulSoup(html, 'html.parser')

        page_num_selector = "#xjs > table > tbody > tr > td:nth-child(" +
                            str(page_num) + ") > a > span"
        driver.find_element_by_css_selector(page_num_selector).click()
        page_num = page_num + 1
        i=1

    is_cnn=soup.select("div.CEMjEf > span")
```

```
# 날짜 받아오기
date=soup.select("p.update-time")

date_re = re.search("₩) ", date[0].text)
date_1=date[0].text[date_re.end():-1]
date_2=dt.datetime.strptime(date_1, '%B %d, %Y')
date_clean = str(date_2.year)+"-"+str(date_2.month)+"-"+str(date_2.day)


# 제목받아오기
title=soup.select("h1")


# 본문 첫번째 문단 받아오기
body_str=""
body_1st = soup.select("#body-text > div.l-container > div.el__leafmedia.

body_re = re.search("(CNN)", body_1st[0].text)
body_clean = body_1st[0].text[body_re.end()+1:]

if body_clean != "":
    body_str = body_str + body_clean

# 본문 두번째 문단 ~ 마지막 문단 받아오기
body_2nd = soup.find_all("div", class_="zn-body__paragraph")
for paragraph_num in range(len(body_2nd)):
    if body_2nd[paragraph_num].text not in  ["", " "]:
        body_str = body_str + body_2nd[paragraph_num].text

# 제목, 날짜, 본문을 리스트에 append
title_list.append(title[0].text)
date_list.append(date_clean)
body_list.append(body_str.replace("  "," "))

# 뒤로가기
driver.back()
```

뉴스 데이터 수집 - 미국 CNN 뉴스

미국 뉴스(CNN) 데이터 크롤링 결과

```
dict_ = {}
dict_["date"] = date_list
dict_["title"] = title_list_list
dict_["text"] = body_list
df = pd.DataFrame(dict_)
df
```

	date	title	text
0	2020-2-26	Biotech company Moderna says its coronavirus v...	Business)US biotech firm Moderna has shipped a...
1	2020-3-1	First death from coronavirus in the United Sta...	A patient infected with the novel coronavirus ...
2	2020-2-22	A controversial religious group is at the cent...	More than half of South Korea's novel coronavi...
3	2020-2-19	Chinese CDC study finds Covid-19 virus to be m...	A comprehensive study of more than 72,000 conf...
4	2020-2-26	Coronavirus has now spread to every continent ...	Public health officials warned Wednesday that ...
5	2020-2-26	CDC official warns Americans it's not a questi...	One of the top officials at the Centers for Di...
6	2020-2-21	China's changed how it counts virus cases thre...	Weeks after the novel coronavirus crisis began...
7	2020-2-14	Over 1,700 frontline medics infected with coro...	Ning Zhu, a nurse in Wuhan, the central Chines...
8	2020-2-17	Did Xi Jinping know about the coronavirus outb...	As the deadly novel coronavirus spread through...
9	2020-2-13	Some coronavirus test kits shipped to states a...	Some of the coronavirus test kits shipped to l...
10	2020-2-29	New coronavirus cases in California and Oregon...	An older adult woman from California is the se...
11	2020-2-27	Top Japanese government adviser says Diamond P...	A top Japanese government adviser has admitted...
12	2020-2-26	San Francisco declares state of emergency over...	San Francisco's mayor on Tuesday declared a lo...
13	2020-2-29	Can Lysol and Clorox products kill the novel c...	Lysol, Clorox and a host of other household di...
14	2020-2-25	Under Trump, America is less prepared for a co...	The coronavirus that emerged from Wuhan, China...

뉴스 데이터 전처리

한국 네이버 뉴스

미국 **CNN** 뉴스

데이터 전처리 - 한국 네이버 뉴스

데이터 읽어오기, 텍스트 정제, 불용어사전

```python
def read_documents(input_file_name):

    corpus = []

    with open(input_file_name, 'rb') as f:
        temp_corpus = pickle.load(f)

    for page in temp_corpus:
        corpus += page

    return corpus
```

```python
def text_cleaning(docs):
    cleaned_docs = []
    for doc in docs:
        temp_doc = re.sub("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "", doc)
        cleaned_docs.append(temp_doc)

    return cleaned_docs
```

한글이 아닌 문자 제거

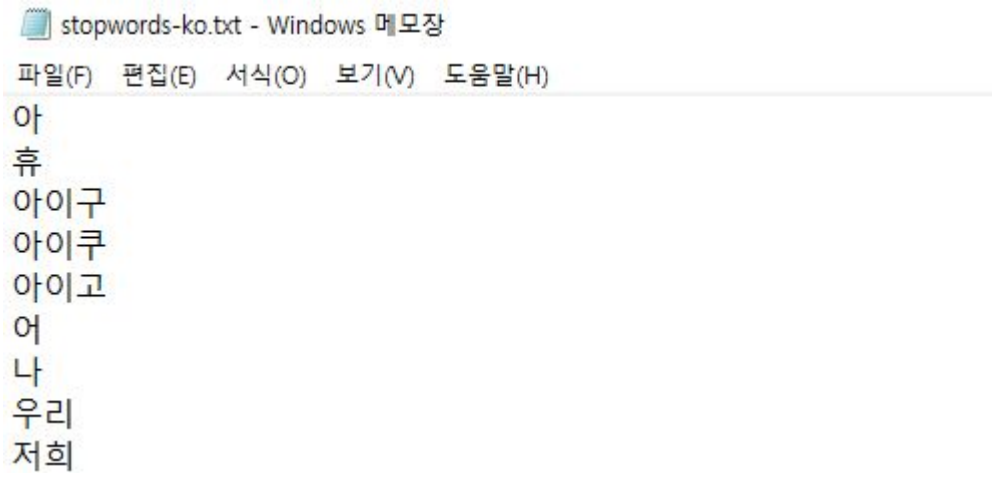
```python
def define_stopwords(path):

    SW = set()
    for i in string.punctuation:
        SW.add(i)

    with open(path, 'r', encoding='utf-8') as f:
        for word in f:
            SW.add(word)

    return SW
```

불용어 목록 만들기

데이터 전처리 - 한국 네이버 뉴스

한국어 불용어 사전

Stopwords-ko.txt



막론하고
관계없이
그치지 않다
그러나
그런데
하지만
든간에
논하지 않다
따지지 않다

그에 따르는
때가 되어
즉
지든지
설령
가령
하더라도
할지라도
일지라도

데이터 전처리 - 한국 네이버 뉴스

Mecab을 사용한 토큰화

```
def mecab_nouns(text):
    nouns = []

    pattern = re.compile('.*₩t[A-Z]+')

    temp =[tuple(pattern.match(token).group(0).split('₩t')) for token in mecab.parse(text).splitlines()[:-1]]
    for token in temp:
        if token[1] == 'NNG' or token[1] == 'NNP' or token[1] == 'NNB' or token[1] == 'NNBC' or token[1] == 'NP' or token[1] == 'NR':
            nouns.append(token[0])

    return nouns

def mecab_morphs(text):
    morphs = []

    pattern = re.compile('.*₩t[A-Z]+')

    temp =[tuple(pattern.match(token).group(0).split('₩t')) for token in mecab.parse(text).splitlines()[:-1]]

    for token in temp:
        morphs.append(token[0])

    return morphs

def mecab_pos(text):
    pos = []
    pattern = re.compile('.*₩t[A-Z]+')
    pos = [tuple(pattern.match(token).group(0).split('₩t')) for token in mecab.parse(text).splitlines()[:-1]]

    return pos
```

형태소

언어학에서 일정한 의미가 있는 가장 작은 말의 단위

(아버지 , NNG)	. . . . .	일반 명사
(가 , JKS)	. . . . .	주격 조사
(방 , NNG)	. . . . .	일반 명사
(에 , JKB)	. . . . .	부사격 조사
(들어가 , VV)	. . . . .	동사
(신다 , EP+EC)	. . . . .	어미

데이터 전처리 - 한국 네이버 뉴스

토큰화 최종 함수

```
def text_tokenizing(corpus, tokenizer):
    token_corpus = []
    if tokenizer == 'noun':
        for n in tqdm_notebook(range(len(corpus)), desc='Preprocessing'):
            token_text = mecab_nouns(corpus[n])
            token_text = [word for word in token_text if word not in SW and len(word) > 1]
            token_corpus.append(token_text)

    elif tokenizer == 'morph':
        for n in tqdm_notebook(range(len(corpus)), desc='Preprocessing'):
            token_text = mecab_morphs(corpus[n])
            token_text = [word for word in token_text if word not in SW and len(word) > 1]
            token_corpus.append(token_text)

    elif tokenizer == 'word':
        for n in tqdm_notebook(range(len(corpus)), desc='Preprocessing'):
            token_text = corpus[n].split()
            token_text = [word for word in token_text if word not in SW and len(word) > 1]
            token_corpus.append(token_text)

    return token_corpus
```

출력 결과

```
print(tokenized_text[3])
```

```
['본문', '내용', '플레이어', '플레이어', '오류', '우회', '위
응', '기금', '올해', '추가', '기여', '아세안', '정상', '대통
착', '위한', '아세안', '지지', '요청', '문재', '대통령', '청
'오후', '화상', '으로', '개최', '아세안', '정상', '회의', '참
께', '극복', '포용', '지속', '가능', '미래', '함께', '만들',
로나', '위기', '보다', '미래', '함께', '만들', '나가', '위한
안', '함께', '위기', '극복', '포용', '지속', '가능', '미래',
'지원', '강화', '시켜', '나갈', '예정', '라고', '일환', '으
'관련', '정상', '회의', '참석', '계기', '남방', '정책', '발표
방', '정책', '단계', '고도', '남방', '정책', '플러스', '발표
안', '국가', '적극', '지지', '협력', '힘입', '아세안', '협력
력', '중요', '부여', '지난', '남방', '정책', '통해', '아세안
로', '백신', '보건', '협력', '강화', '통해', '아세안', '코로
'대해', '사의', '표했', '또한', '아세안', '정상', '으로', '한
력', '지속', '가능', '미래', '함께', '준비', '나가', '자고',
력', '강화', '대한', '의지', '확인', '공동', '성명', '채택',
'평화', '번영', '역내', '질서', '유지', '발전', '시켜', '나가
'대화', '조속', '재개', '통해', '한반도', '완전', '비핵화',
'정상회의', '그간', '남방', '정책', '기반', '아세안', '협력'
'밝혔', '본문', '내용']
```

데이터 전처리 - 미국 CNN 뉴스

데이터 읽어오기

```
def read_texts(df):
    texts=[]
    for i in range(len(df)):
        words =  WordPunctTokenizer().tokenize(df["text"][i])
        texts.append(words)
    print(texts)
    return texts
```

	date	title	text
0	2021-10-15	FDA vaccine advisers recommend emergency use a...	Vaccine advisers to the US Food and Drug Admin...
1	2021-10-8	Studies confirm waning immunity from Pfizer's ...	Two real-world studies published Wednesday con...
2	2021-10-8	Here's what having a Covid-19 vaccine for chil...	Pfizer said Thursday it's asked the US Food an...
3	2021-10-12	Texas governor bans Covid-19 vaccine mandates ...	Texas Gov. Greg Abbott on Monday issued an exe...
4	2021-10-4	NYC vaccine mandate takes effect with 96% of t...	New York City Mayor Bill de Blasio said 96% of...
...	...	...	...
1045	2020-2-8	New study an eye-opener on how coronavirus is ...	A study published Friday in the medical journa...
1046	2020-2-3	A soldier surprised his mom as she was sworn i...	Erika Benning's heart was already racing as sh...
1047	2020-2-19	An American evacuated from Japan on a US chart...	An American who was evacuated on a US-chartere...
1048	2020-2-15	US to evacuate Americans on cruise ship quaran...	The US government is preparing to evacuate Ame...
1049	2020-2-11	One man linked to at least 9 coronavirus cases...	A British man linked to a number of coronaviru...

크롤링한 뉴스 기사
파일 읽어오기

데이터 전처리 - 미국 CNN 뉴스

NLTK를 사용한 토큰화, 텍스트 정제

```
def tokenize_pos_tag(df):
    res=[]
    for i in range(len(df)):
        sentence = df["text"][i]
        tokenize_pos = pos_tag(WordPunctTokenizer().tokenize(sentence))
        res.append(tokenize_pos)
    return res


def ANRV(words_pos):
    res = []
    for i in range(len(words_pos)):
        tmp=[]
        for j in range(len(words_pos[i])):
            if words_pos[i][j][1][0] in ["J","R","V"]:
                tmp.append((words_pos[i][j][0].lower(), words_pos[i][j][1][0].lower()))
            elif words_pos[i][j][1][0] == "N":
                if words_pos[i][j][1] == "NNP" or words_pos[i][j][1] == "NNPS":
                    tmp.append((words_pos[i][j][0].lower(), words_pos[i][j][1][0].lower()))
                else:
                    tmp.append((words_pos[i][j][0].lower(), words_pos[i][j][1][0].lower()))
        res.append(tmp)
    return res
```

품사 태깅

def tokenize_pos_tag

'Vaccine advisers to the US Food and Drug Administration voted

```
[('Vaccine', 'NN'),
 ('advisers', 'NNS'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('US', 'NNP'),
 ('Food', 'NNP'),
 ('and', 'CC'),
 ('Drug', 'NNP'),
 ('Administration', 'NNP'),
 ('voted', 'VBD')]
```

def ANRV

```
[('Vaccine', 'NN'),
 ('advisers', 'NNS'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('US', 'NNP'),
 ('Food', 'NNP'),
 ('and', 'CC'),
 ('Drug', 'NNP'),
 ('Administration', 'NNP'),
 ('voted', 'VBD')]
```

```
[('vaccine', 'n'),
 ('advisers', 'n'),
 ('us', 'n'),
 ('food', 'n'),
 ('drug', 'n'),
 ('administration', 'n'),
 ('voted', 'v'),
 ('unanimously', 'r'),
 ('thursday', 'n'),
 ('recommend', 'v')]
```

데이터 전처리 - 미국 CNN 뉴스

NLTK를 사용한 텍스트 정제, 불용어 제거

```
def Lemma(words_anrv):
    lm = WordNetLemmatizer()
    res=[]
    for i in range(len(words_anrv)):
        tmp=[]
        for j in range(len(words_anrv[i])):
            #print(words_anrv[i][j][0])
            if words_anrv[i][j][1] == "j":
                pos = "a"
            else:
                pos=words_anrv[i][j][1]
            #print(lm.lemmatize(words_anrv[i][j][0], pos=pos))
            tmp.append(lm.lemmatize(words_anrv[i][j][0], pos=pos))
        res.append(tmp)
    return res


def clean_stopword(words):
    stop_words = stopwords.words("english")
    res = []
    for i in range(len(words)):
        res.append([w for w in words[i] if w not in stop_words and len(w) > 3])
    return res
```

원형 복원 / 불용어 제거

def Lemma

```
[('vaccine', 'n'),
 ('advisers', 'n'),
 ('us', 'n'),
 ('food', 'n'),
 ('drug', 'n'),
 ('administration', 'n'),
 ('voted', 'v'),
 ('unanimously', 'r'),
 ('thursday', 'n'),
 ('recommend', 'v')]
```

→

```
['vaccine',
 'adviser',
 'u',
 'food',
 'drug',
 'administration',
 'vote',
 'unanimously',
 'thursday',
 'recommend']
```

def clean_stopword

```
['vaccine',
 'adviser',
 'u',
 'food',
 'drug',
 'administration',
 'vote',
 'unanimously',
 'thursday',
 'recommend']
```

→

```
['vaccine',
 'adviser',
 'food',
 'drug',
 'administration',
 'vote',
 'unanimously',
 'thursday',
 'recommend',
 'emergency']
```

데이터 전처리 - 미국 CNN 뉴스

미국 뉴스 데이터 전처리

```
news_list = []
tokened_news = []
for i in range(len(news_main_texts)):
    for j in range(len(news_main_texts[i])):
        news_list = news_main_texts[i][j]
        text_pp = tokenize_text2(news_list)
        tokened_news.append(text_pp)
print(tokened_news)
```

출력 결과

```
[[['business', 'biotech', 'firm', 'moderna', 'ship', 'experimental', 'coronavirus', 'vacci
art', 'work', 'immunization'], ['initial', 'trial', 'potential', 'vaccine', 'begin', 'apri
'least', 'year', 'moderna', 'mrna', 'say', 'statement', 'monday', 'first', 'batch', 'novel
'send', 'national', 'institute', 'allergy', 'infectious', 'disease', 'niaid', 'share', 'co
h', 'new', 'york', 'tuesday', 'moderna', 'say', 'first', 'vial', 'experimental', 'vaccine'
d', 'state', 'typically', 'involve', 'test', 'vaccine', 'small', 'number', 'healthy', 'hum
i', 'say', 'clinical', 'trial', 'start', 'end', 'april', 'first', 'step', 'potentially', '
l', 'street', 'journal', 'first', 'report', 'development', 'say', 'dos', 'vaccine', 'test'
'response', 'protect', 'virus'], ['fauci', 'tell', 'cnn', 'people', 'participate', 'trial'
'test', 'regulatory', 'approval', 'need', 'vaccine', 'deploy', 'widely'], ['health', 'offi
'work', 'breakneck', 'pace', 'identify', 'treatment', 'vaccine', 'help', 'fight', 'coronav
'previously', 'tell', 'cnn', 'researcher', 'expedite', 'approval', 'process', 'vaccine', '
tempt', 'halt', 'spread', 'virus', 'even', 'proceed', 'emergency', 'speed', 'vaccine', 'av
```

워드 클라우드

한국 네이버 뉴스

미국 **CNN** 뉴스

워드 클라우드 - 한국 네이버 뉴스

워드 클라우드 코드

```
# 트위터에서 만든 소셜 분석을 위한 형태소 분석기 Okt 사용
okt = Okt()
myList = okt.pos(content_list, norm=True, stem=True) # 모든 형태소 추출
myList_filter = [x for x, y in myList if y in ['Noun']] # 추출된 값 중 명사만 추출

Okt = Text(myList_filter, name="Okt")

# 그래프에서 한글이 출력이 안되는 문제 해결 (ㅁㅁㅁ 처럼 출력됨)
font_location = "c:/Windows/Fonts/malgun.ttf"
font_name = font_manager.FontProperties(fname=font_location).get_name()
rc('font', family=font_name)

# 그래프 x, y 라벨 설정
plt.xlabel("명사")
plt.ylabel("빈도")

# 그래프에서 x, y 값을 설정
wordInfo = dict()
for tags, counts in Okt.vocab().most_common(50):
    if(len(str(tags)) > 1):
        wordInfo[tags] = counts

values = sorted(wordInfo.values(), reverse=True)
keys = sorted(wordInfo, key=wordInfo.get, reverse=True)

# 그래프 값 설정
plt.figure(figsize=(15,10))
plt.bar(range(len(wordInfo)), values, align='center')
plt.xticks(range(len(wordInfo)), list(keys), rotation='70')
plt.show()


# wordCloud 출력
plt.figure(figsize=(15,10))
wc = WordCloud(width = 1000, height = 600, background_color="white", font_path=font_location, max_words=50)
plt.imshow(wc.generate_from_frequencies(Okt.vocab()))
plt.axis("off")
plt.show()
```

트위터 (Okt)

twitter/**twitter-korean-text**

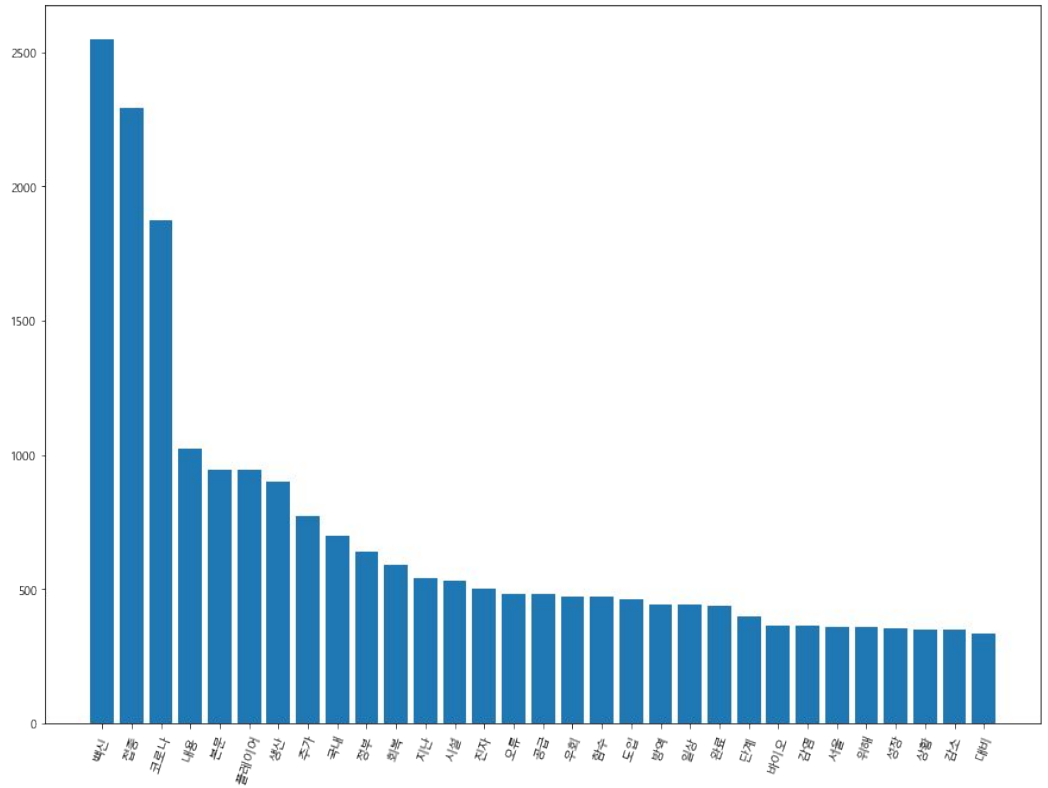
Korean tokenizer

13 Contributors | 145 Used by | 781 Stars | 159 Forks

형태소 분석기를 활용하여(명사) 단어 별 토큰화 진행 -> wordcloud

워드 클라우드 - 한국 네이버 뉴스

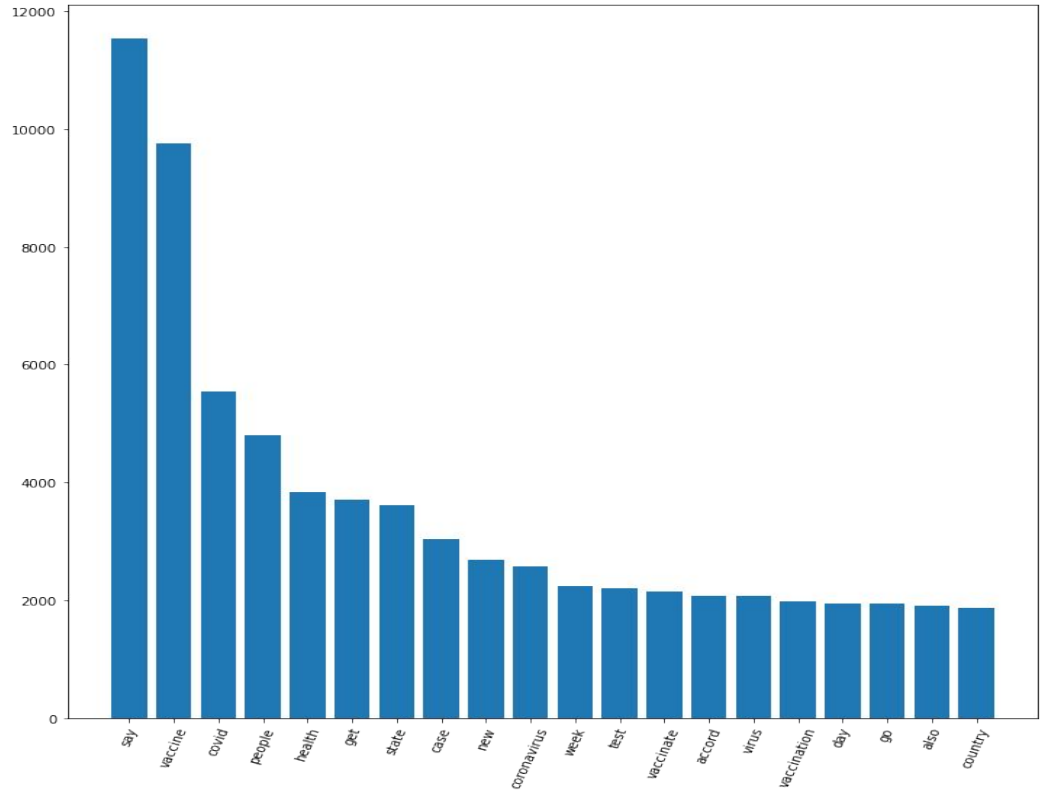
뉴스 단어 빈도수



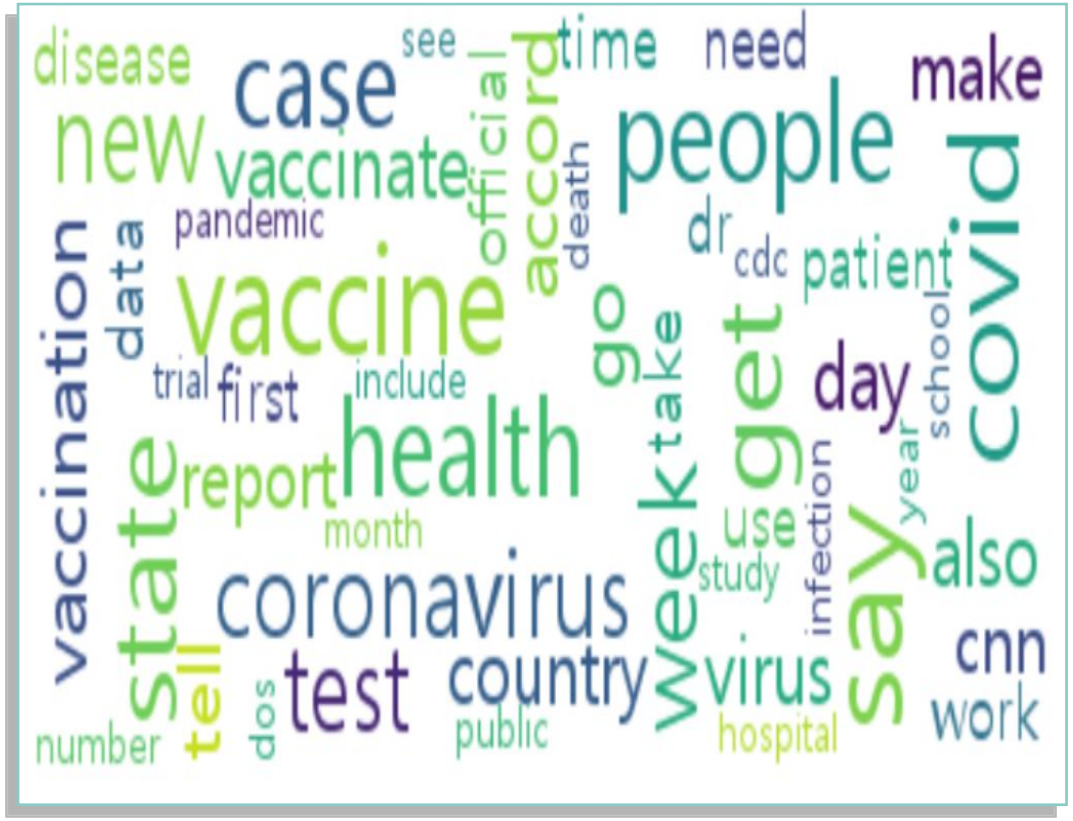
뉴스 워드 클라우드

워드 클라우드 - 미국 CNN 뉴스

미국 뉴스 단어 빈도수



미국 뉴스 워드 클라우드



토픽 모델링

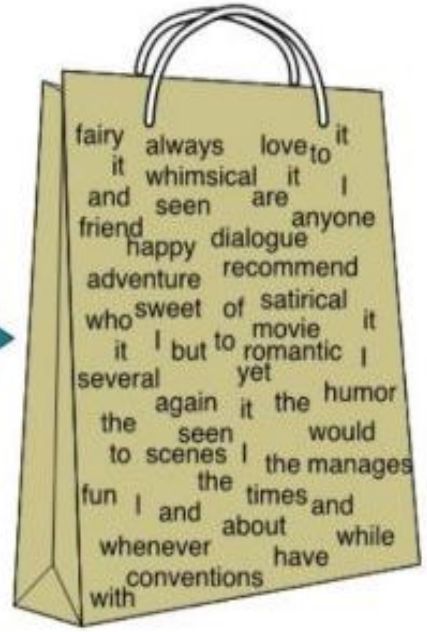
한국 네이버 뉴스

미국 **CNN** 뉴스

토픽 모델링

Gensim

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Bag of words

Bag of Words란

단어들의 순서는 고려하지 않고,

단어들의 출현 빈도에만 집중하는

텍스트 데이터의 수치화 표현 방법

해당 문서 내에서

특정 단어가 N번 등장했다면,

이 가방에는 그 특정 단어가 N개 있게 됨

토픽 모델링 - 한국 네이버 뉴스

Code

함수화 적용

```
input_file_name = "C:/Users/kazio/Downloads/CCCC/k_code/naver_news_content.pk"
documents = read_documents(input_file_name)
SW = define_stopwords("C:/Users/kazio/Downloads/CCCC/stopwords-ko.txt")
cleaned_text = text_cleaning(documents)
tokenized_text = text_tokenizing(cleaned_text, tokenizer="morph")
```

딕셔너리 / 코퍼스

```
dictionary = corpora.Dictionary(tokenized_text)
corpus = [dictionary.doc2bow(text) for text in tokenized_text]
```

```
print(dictionary)
```

```
Dictionary(8550 unique tokens: ['가능', '가장', '각국', '감사', '강조']...)
```

토픽 모델링 - 한국 네이버 뉴스

TF-IDF

TF-IDF(Term Frequency-Inverse Document Frequency):
단어의 빈도와 역 문서 빈도(문서의 빈도에 특정 식을 취함)를 사용하여 DTM 내의 각 단어들마다 중요한 정도를 가중치로 주는 방법

$$idf(d,t) = log(\frac{n}{1+df(t)})$$

TF-IDF는 **모든 문서**에서 **자주 등장**하는 단어는
중요도가 낮다고 판단하며, ex) a, the
특정 문서에서만 **자주 등장**하는 단어는
중요도가 높다고 판단.
TF-IDF 값이 낮으면 중요도가 낮은 것이며
TF-IDF 값이 크면 중요도가 큰 것.

TF-IDF Model

```
tfidf = models.TfidfModel(corpus)
corpus_tfidf = tfidf[corpus]
corpus_tfidf[2][:5]
```

```
[(0, 0.018916727108076122),
 (1, 0.020354589699820753),
 (2, 0.031224685165735507),
 (3, 0.029297937295090025),
 (4, 0.038274061043506 58)]
```

토픽 모델링

Lda Model

Latent Dirichlet Allocation

「잠재 디리클레 할당」

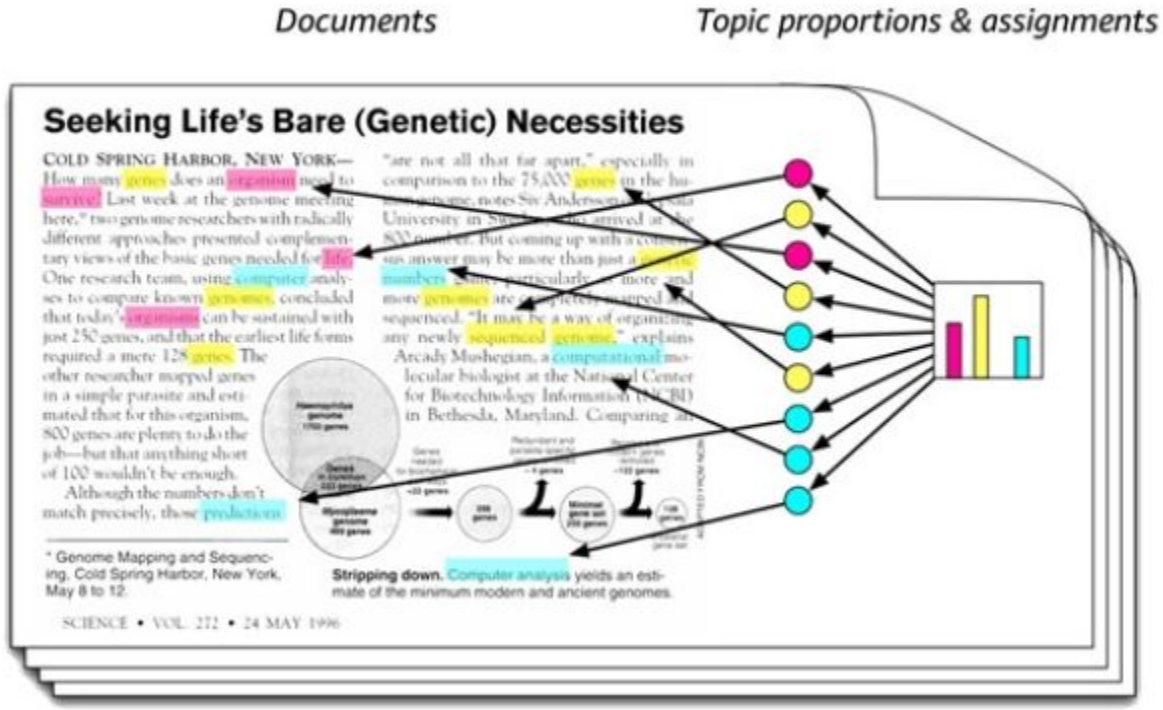
LDA를 토픽모델링(Topic Modeling) 기법이라고 부르는데
단어나 문서의 숨겨진 주제(Topic)을 찾아내 주기 때문.

여기서 Topic(주제)란, 수집된 원문 내용에 담긴 다양한
키워드를 기반으로 내용을 유형화(그룹화) 시켜주는 것

추출한 원문에는 다양한 내용이 담겨 있을 수 있는데
이러한 주제들을 일일이 수작업으로 분류하기 어렵기 때문에
LDA 같은 분류 방법을 적용해 전반적인 데이터의 구조를
먼저 파악하는 것이 중요

LDA 기법은 단순히 주제만 분류해주는 것이 아니라
주제에 포함되는 키워드들을 보여주기 때문에
그 키워드들로 해당 주제를 해석하고 정의할 수 있음

물론, 사전에 충분한 데이터의 정제 과정이 요구되며
기계적 분류인 만큼 결과 자체가 완전하지 않을 수도 있어
어느 정도의 후보정이 필요할 수도 있음



토픽 모델링 - 한국 네이버 뉴스

TF-IDF

```
# 토픽 개수, 키워드 개수를 정해주는 변수를 추가.
NUM_TOPICS = 3

NUM_TOPIC_WORDS = 30

def print_topic_words(model): # model = LDA된 결과

    # 토픽 모델링 결과를 출력해 주는 함수.
    print("₩nPrinting topic words.₩n")

    for topic_id in range(model.num_topics):
        topic_word_probs = model.show_topic(topic_id, NUM_TOPIC_WORDS)
        print('Topic ID: {}'.format(topic_id))

        for topic_word, prob in topic_word_probs:
            print('₩t{}₩t{}'.format(topic_word, prob))
```

```
dictionary = corpora.Dictionary(tokenized_text)
corpus = [dictionary.doc2bow(text) for text in tokenized_text]
```

```
print(dictionary)
```

```
Dictionary(8550 unique tokens: ['가능', '가장', '각국', '감사', '강조']...)
```

```
# document-term matrix를 만들고,
corpus, dictionary = build_doc_term_mat(tokenized_text)
# LDA를 실행.
model = models.ldamodel.LdaModel(corpus_tfidf, num_topics=NUM_TOPICS, id2word=dictionary, alpha='auto', eta='auto')
```

```
model.show_topic(0, 10)
```

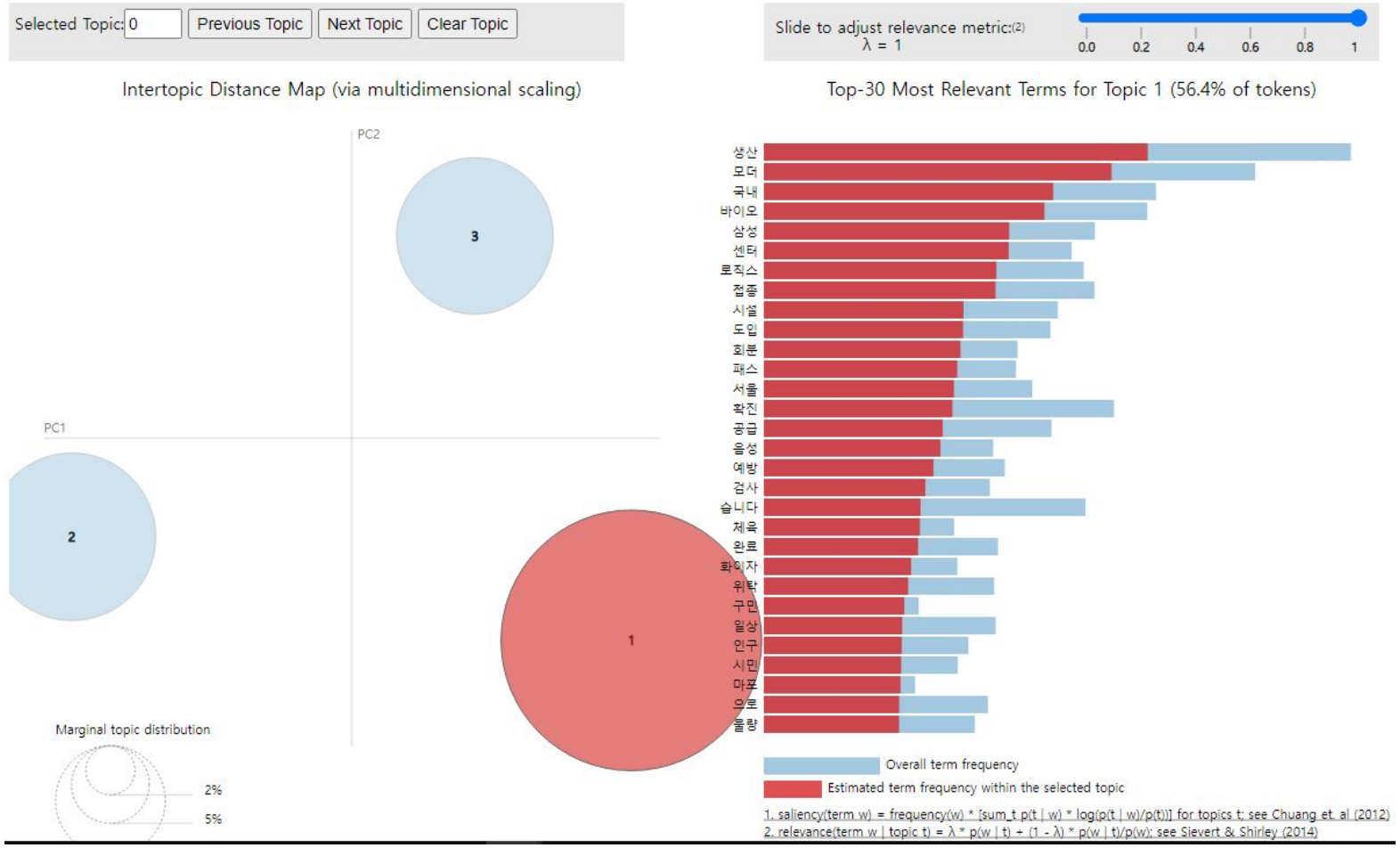
```
[('분기', 0.0027425492),
 ('생산', 0.0023713612),
 ('모더', 0.0015388113),
 ('국내', 0.0014743245),
 ('판매', 0.0014681031),
 ('수출', 0.0014547467),
 ('성장', 0.0014490802),
 ('시설', 0.0013182973),
 ('공급', 0.0012509981),
 ('패스', 0.0012350699)]
```

토픽 모델링 - 한국 네이버 뉴스

pyLDAvis

```
# pyLDAvis 불러오기
import pyLDAvis
import pyLDAvis.gensim
# pyLDAvis를 jupyter notebook에서 실행할 수 있게 활성화.
pyLDAvis.enable_notebook()

# pyLDAvis 실행.
data = pyLDAvis.gensim.prepare(model, corpus, dictionary)
data # print X 그냥 실행
```

출력 결과



토픽 모델링 - 미국 CNN 뉴스

토픽 모델링

```
dictionary = corpora.Dictionary(clean_words)
corpus = [dictionary.doc2bow(text) for text in clean_words]
print(dictionary)
```

```
Dictionary(16733 unique tokens: ['administer', 'administration', 'advancing', 'adviser', 'advisory']...)
```

딕셔너리 / 코퍼스

```
tfidf = models.TfidfModel(corpus)
corpus_tfidf = tfidf[corpus]
```

TF-IDF Model

```
model = models.ldamodel.LdaModel(corpus_tfidf, num_topics=2, id2word=dictionary, alpha='auto', eta='auto')
```

```
model.show_topic(0, 10)
```

```
[('vaccine', 0.0013949742),
 ('trial', 0.0009652185),
 ('variant', 0.00084375875),
 ('county', 0.0008188072),
 ('vaccination', 0.00077285676),
 ('patient', 0.0007213663),
 ('trump', 0.00071629393),
 ('johnson', 0.0007124798),
 ('school', 0.00070121395),
 ('case', 0.0006736684)]
```

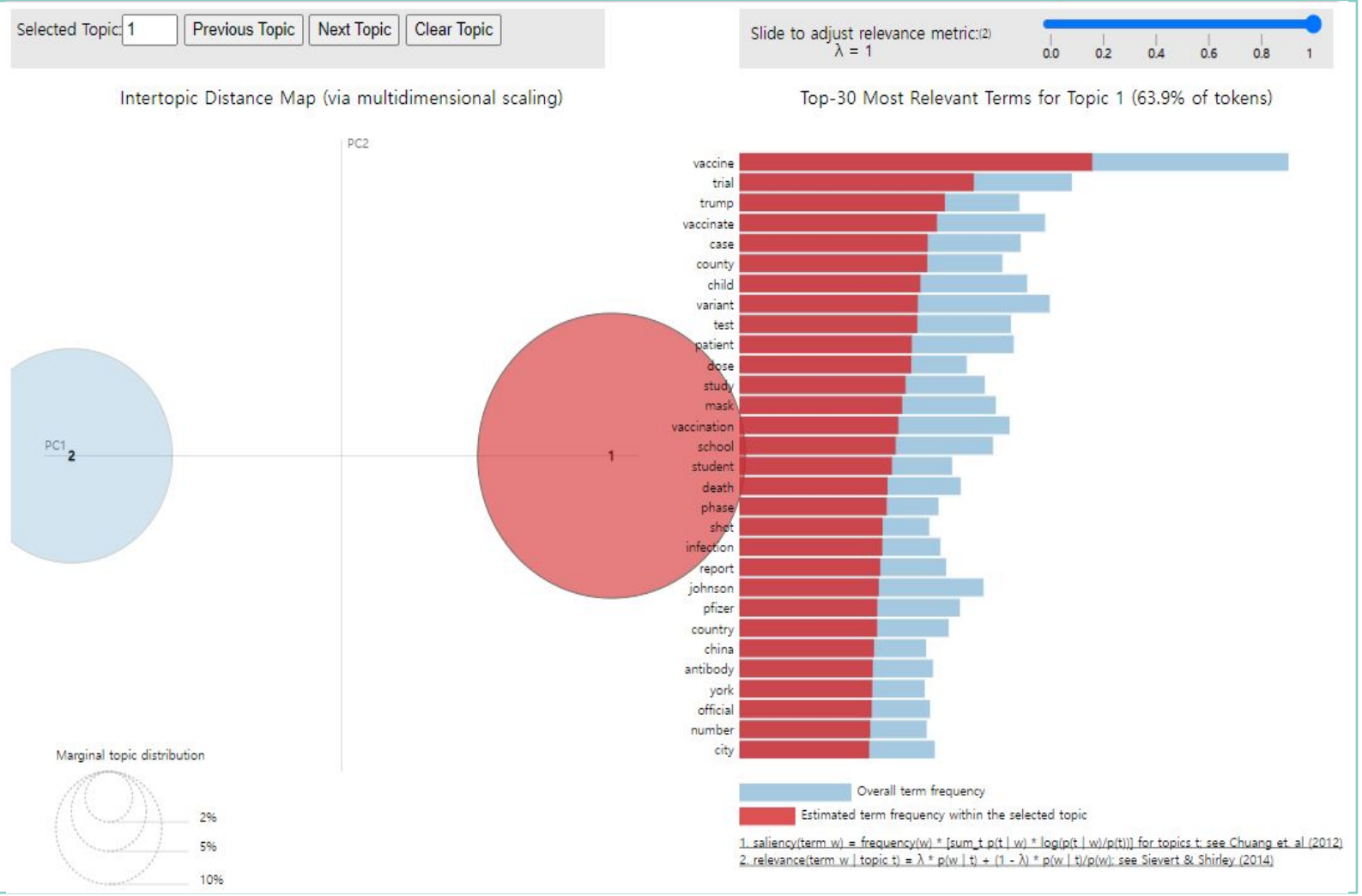
LDA 모델 적용

토픽 모델링 - 미국 CNN 뉴스

토픽 모델링

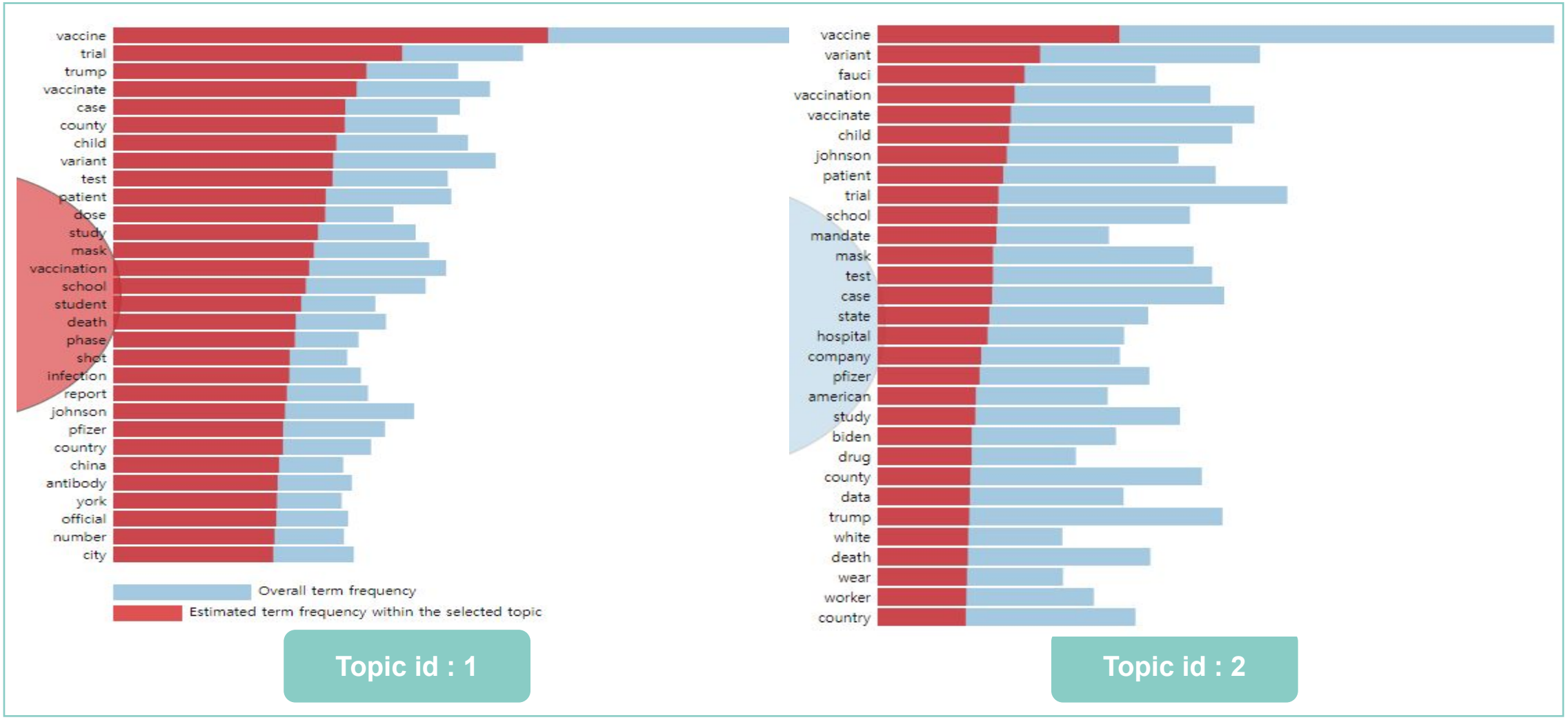
```
# pyLDAvis 불러오기
import pyLDAvis
import pyLDAvis.gensim
# pyLDAvis를 jupyter notebook에서 실행할 수 있게 활성화.
pyLDAvis.enable_notebook()

# pyLDAvis 실행.
data = pyLDAvis.gensim.prepare(model, corpus, dictionary)
data # print X 그냥 실행
```

출력 결과



토픽 모델링 - 미국 CNN 뉴스

토픽 모델링



네이버 블로그

데이터 수집

워드 클라우드

토픽 모델링

블로그 데이터 수집 - 네이버 블로그

네이버 블로그 api 활용

```
# 크롬 웹브라우저 실행
path = "f:/chromedriver.exe"

driver = webdriver.Chrome(path)
url_list = []
content_list = ""
text = "코로나 백신"

for i in range(1, 100):  # 1~100페이지까지의 블로그 내용을 읽어옴
    url = 'https://section.blog.naver.com/Search/Post.nhn?pageNo='+ str(i) + '&rangeType=ALL&orderBy=sim&keyword=' + text
    driver.get(url)
    time.sleep(0.5)

    for j in range(1, 7): # 각 블로그 주소 저장
        try:
            titles = driver.find_element_by_xpath('/html/body/ui-view/div/main/div/div/section/div[2]/div['+str(j)+']/div/div[1]/div[1]/a[
            title = titles.get_attribute('href')
        except:
            title = "OUTLINK"

        url_list.append(title)

print("url 수집 끝, 해당 url 데이터 크롤링")

for url in url_list: # 수집한 url 만큼 반복
    driver.get(url) # 해당 url로 이동

    driver.switch_to.frame('mainFrame')
    overlays = ".se-component.se-text.se-l-default" # 내용 크롤링
    contents = driver.find_elements_by_css_selector(overlays)

    for content in contents:
        content_list = content_list + content.text # content_list 라는 값에 + 하면서 점점 누적
```

출력 결과

```
print(content_list)
type(content_list)
```

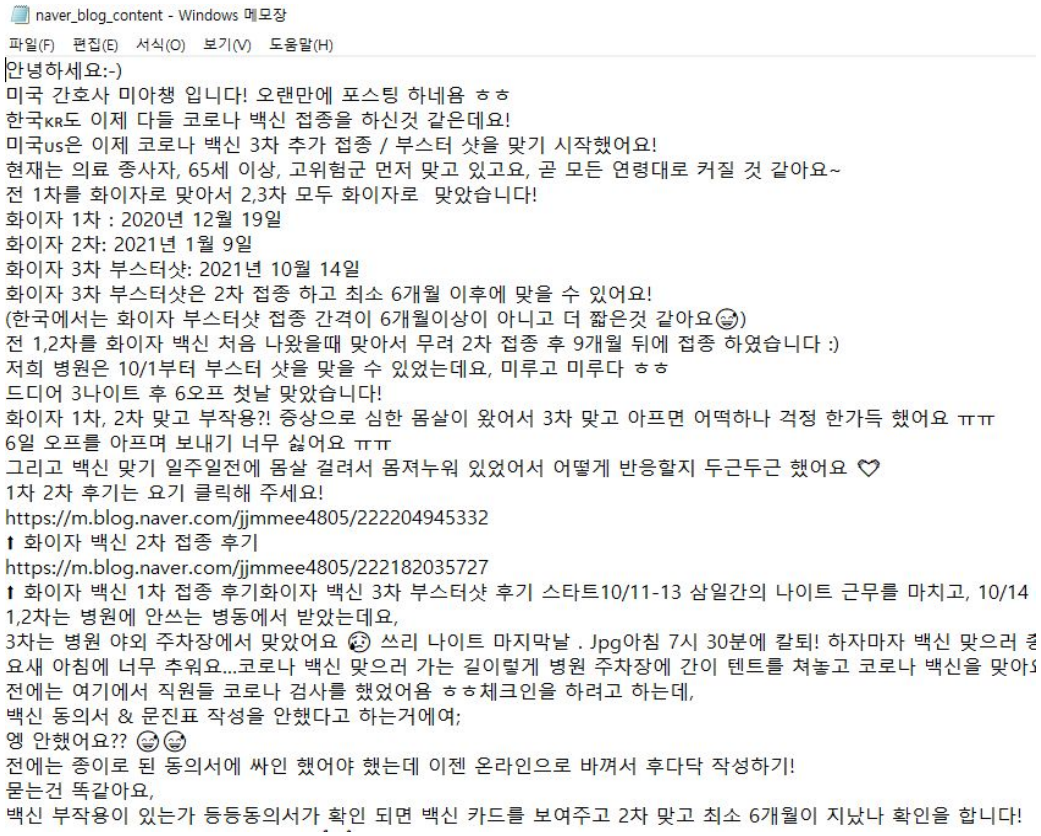
```
안녕하세요:-)
미국 간호사 미아챙 입니다! 오랜만에 포스팅 하네욤 ㅎㅎ
한국KR도 이제 다들 코로나 백신 접종을 하신것 같은데요!
미국US은 이제 코로나 백신 3차 추가 접종 / 부스터 샷을 맞기 시작했어요!
현재는 의료 종사자, 65세 이상, 고위험군 먼저 맞고 있고요, 곧 모든 연령대로 커질 것 같아요~
전 1차를 화이자로 맞아서 2,3차 모두 화이자로  맞았습니다!
화이자 1차 : 2020년 12월 19일
화이자 2차: 2021년 1월 9일
화이자 3차 부스터샷: 2021년 10월 14일
화이자 3차 부스터샷은 2차 접종 하고 최소 6개월 이후에 맞을 수 있어요!
(한국에서는 화이자 부스터샷 접종 간격이 6개월이상이 아니고 더 짧은것 같아요😅)
전 1,2차를 화이자 백신 처음 나왔을때 맞아서 무려 2차 접종 후 9개월 뒤에 접종 하였습니다 :)
저희 병원은 10/1부터 부스터 샷을 맞을 수 있었는데요, 미루고 미루다 ㅎㅎ
드디어 3나이트 후 6오프 첫날 맞았습니다!
화이자 1차, 2차 맞고 부작용?! 증상으로 심한 몸살이 왔어서 3차 맞고 아프면 어떡하나 걱정 한가득 했
6일 오프를 아프며 보내기 너무 싫어요 ㅠㅠ
그리고 백신 맞기 일주일전에 몸살 걸려서 몸져누워 있었어서 어떻게 반응할지 두근두근 했어요 ❤️
```

셀레니움을 활용하여 네이버 블로그 본문 내용 가져옴 (출력 결과 확인)

블로그 데이터 수집 - 네이버 블로그

블로그 데이터 저장

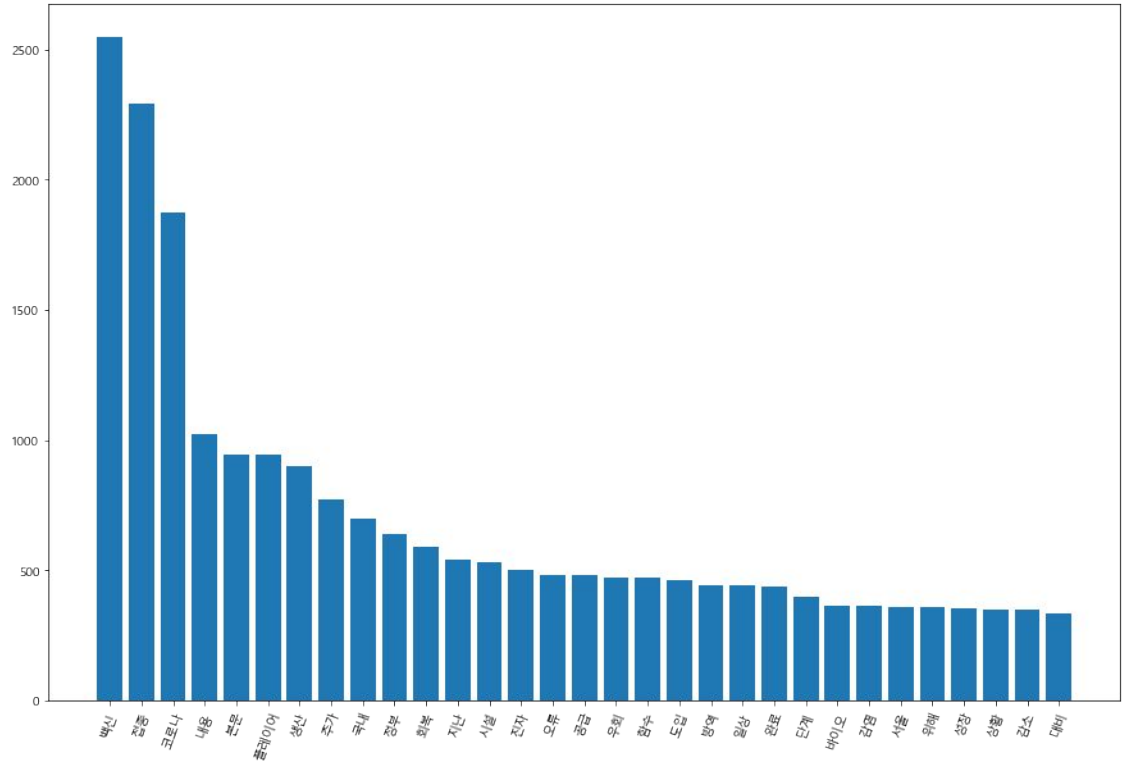
```
## txt 파일로 저장
# content_list
f = open('naver_blog_content.txt', 'w', encoding='utf-8')
f.write(content_list)
f.close()
```



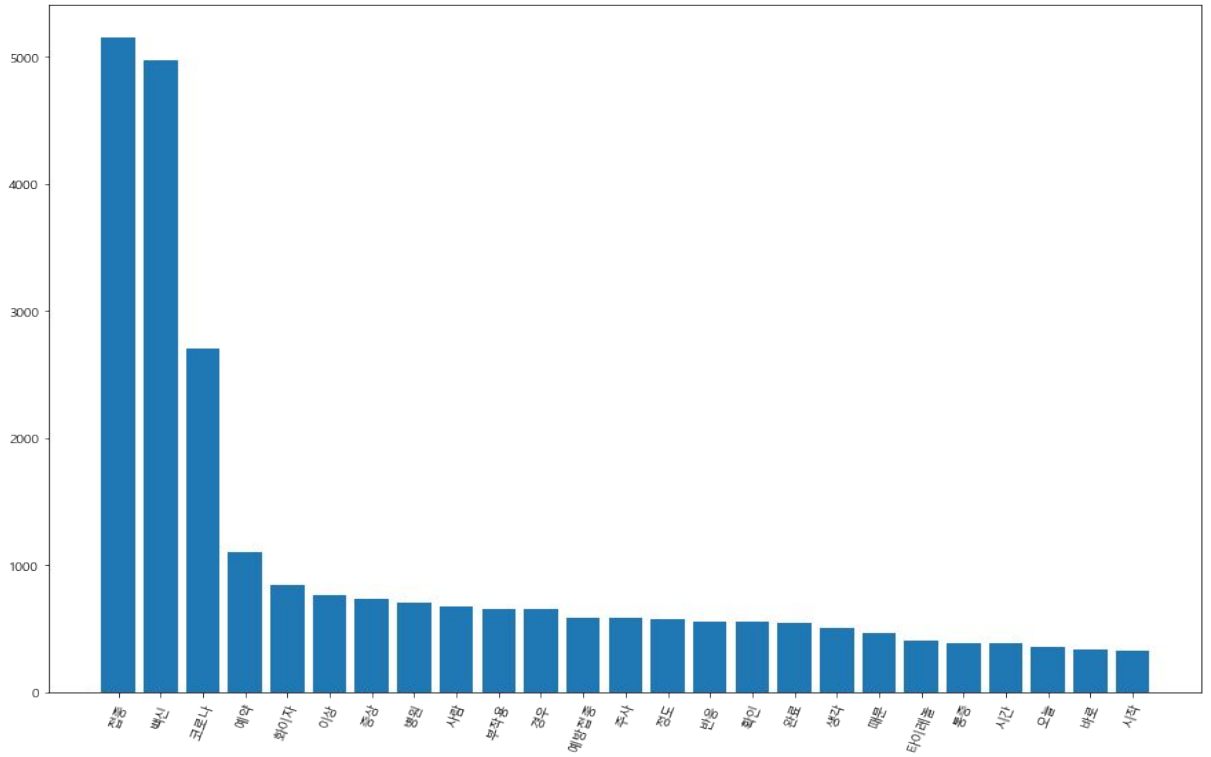
블로그 본문 내용을 모두 하나의 **str**로 만든 후 **txt**파일로 저장

뉴스 / 블로그 비교 - 단어 빈도수

뉴스 단어 빈도수



블로그 단어 빈도수



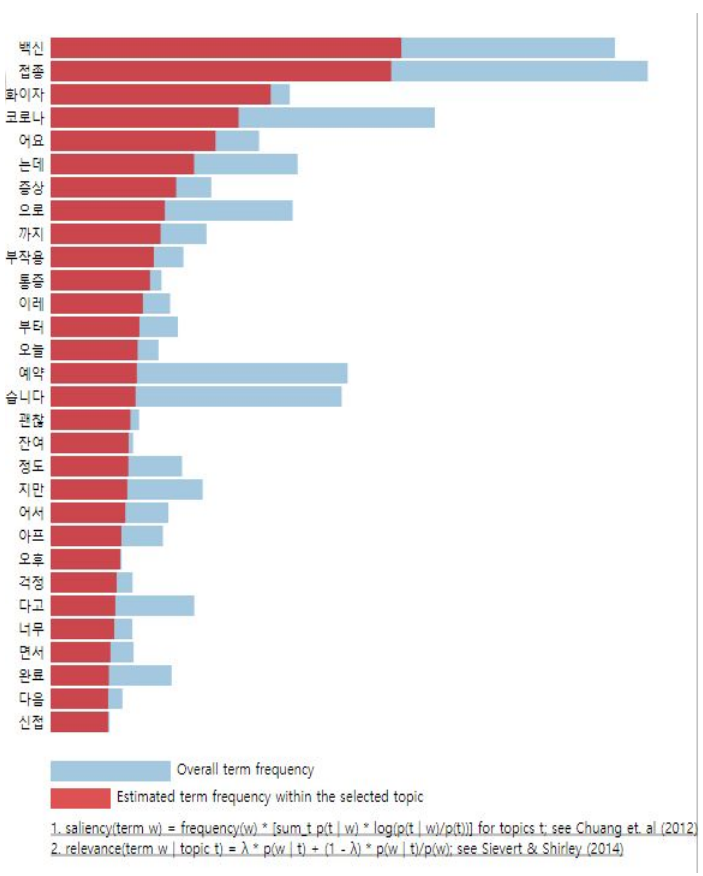
뉴스 / 블로그 비교 - 워드 클라우드

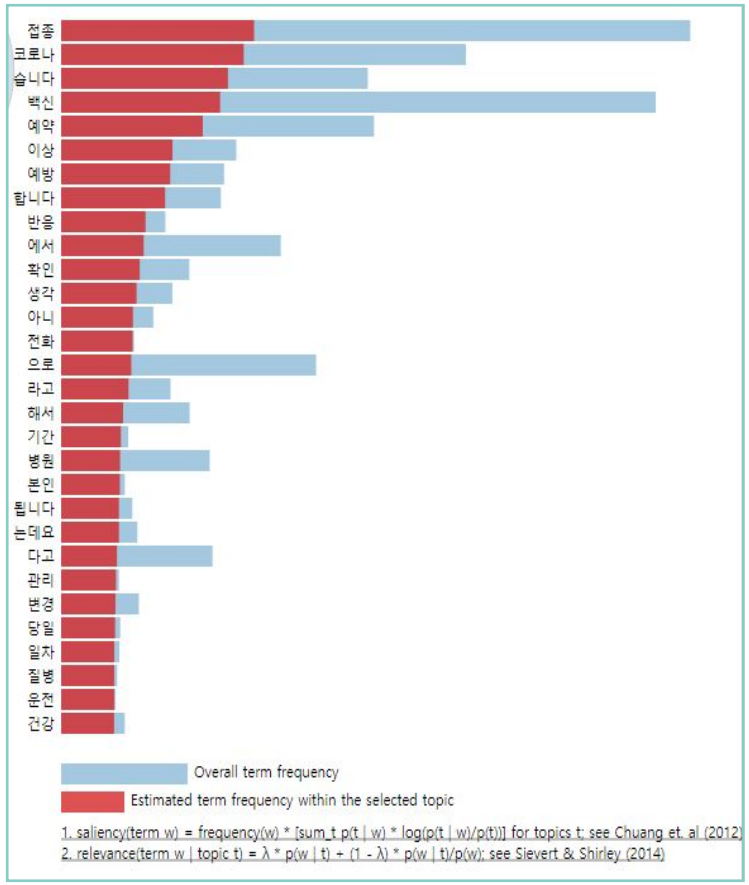
블로그 워드 클라우드

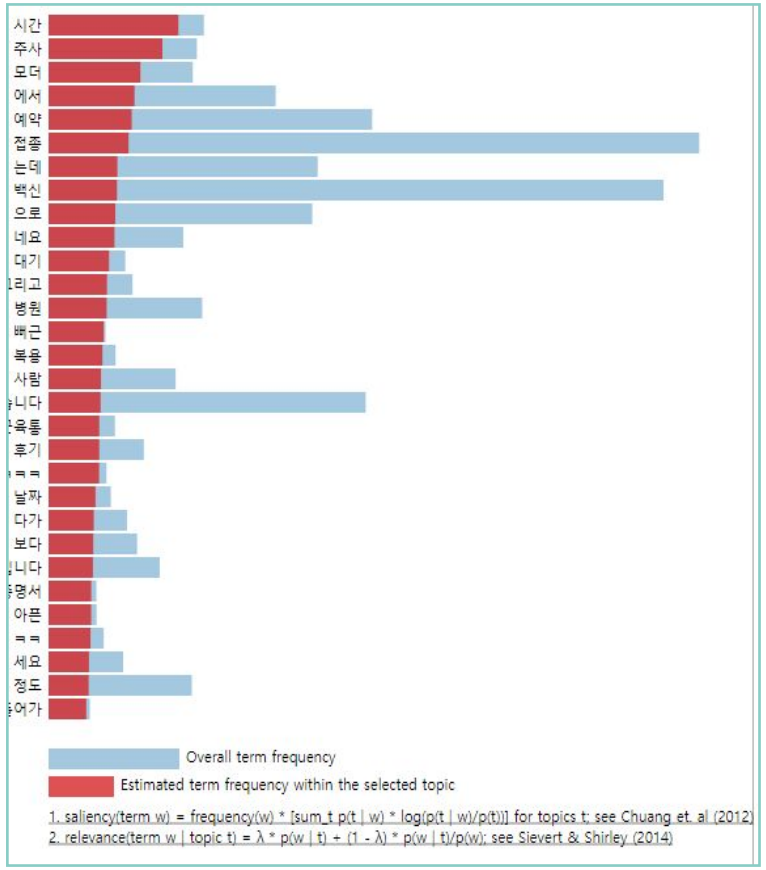
뉴스 워드 클라우드

토픽 모델링 - 네이버 블로그

pyLDAvis







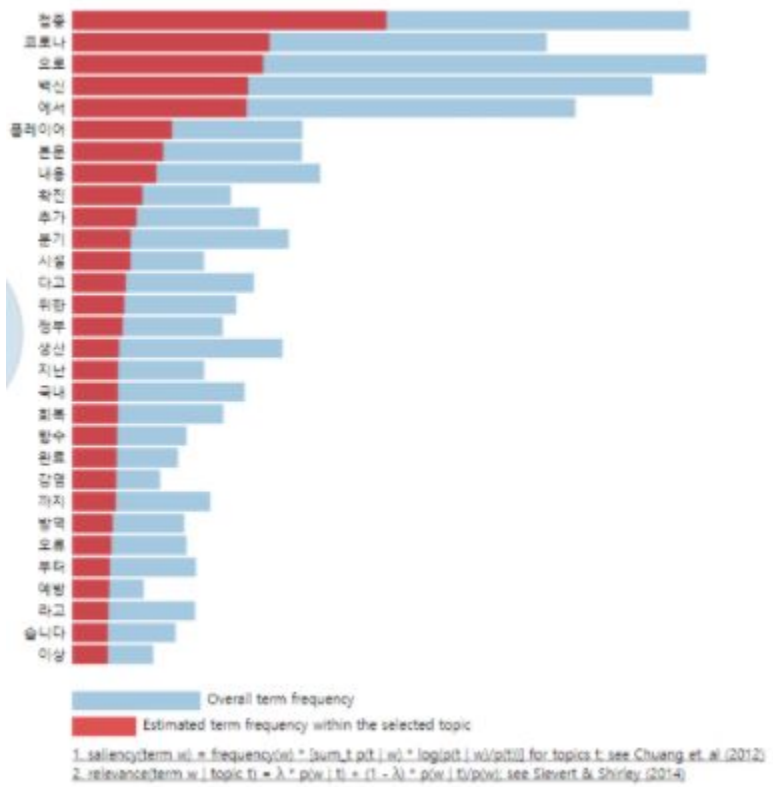
Topic id : 0

Topic id : 1

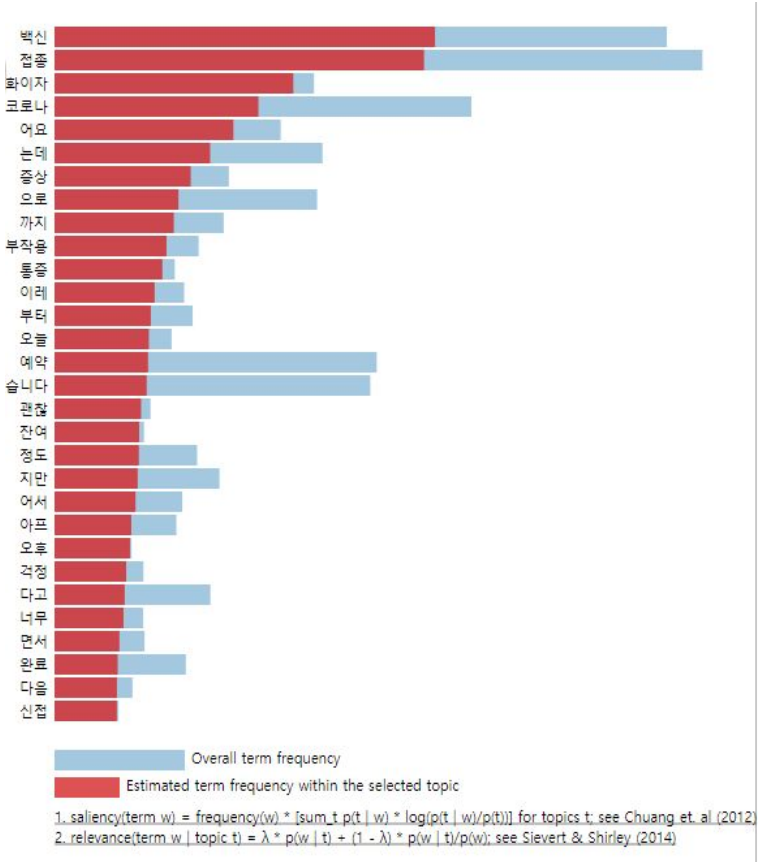
Topic id : 2

뉴스 / 블로그 비교 - 토픽 모델링

뉴스 토픽 모델링



블로그 토픽 모델링



느낀점 & 한계점

역할	느낀점	한계점
정연규 국내 데이터 수집, 워드클라우드, ppt 작성 **정소비** 미국 데이터 수집, CNN 데이터 전처리, 발표 **정현석** 데이터 전처리, 데이터 토큰화, 토픽모델링	• 한번도 접해보지 못한 자연어 처리라는 주제를 선정하면서 자연어 모델에 알맞게 텍스트를 전처리하는 방법에 대해 배울 수 있었다. • 자연어 처리를 하려니 생각보다 공부해야 할 부분이 많았다 • konlpy 설치부터 자연어 처리에 모든 것이 처음이였지만 크롤링부터 토픽 모델링까지 나름 의미있는 결과를 도출해서 많이 배우기도 하고 좋았다.	• 국내는 네이버 api를 사용하여 데이터 크롤링를 하였지만 미국같은 경우 언론 보도 글이 아닌 이상 데이터를 수집하는 데 어려움을 느꼈습니다. • 영화평점이나 쇼핑몰 리뷰같이 라벨이 있는 데이터가 아니라서 이를 모델에 넣고 검증할 수 있는 데엔 한계를 느꼈습니다.

Q&A