네이버 뉴스 크롤링을 이용한 형태소 분석

홍혜린

네이버 뉴스 크롤링을 이용한 형태소 분석과 시각화

- 매일 접하는 네이버 뉴스를 크롤링해 자주 나오는 단어를 찾고, 형태소 분석을 통해 특정 품사만을 채택합니다.
- 이후 Word Cloud를 통해 시각화를 진행합니다.

Komoran을 이용한 한국어 형태소 분석

Komoran - (KOrean MORphological ANalyzer)

- 자바로 구현된 형태소 분석이므로 환경설정을 맞추고 진행을 합니다.
- Beautiful Soup (bs4)를 이용해 네이버 뉴스를 크롤링합니다.

```
from konlpy.tag import Mecab, Komoran
```

```
tokenizer = Komoran()
```

```
def crawling_news(category, from_, to_):
    total = pd.DataFrame()
    base_url = "https://news.naver.com/main/list.nhn?"
    para = {'mode': ['LS2D'],
            'mid': ['shm'],
            'sid2': ['259'],
            'sid1': ['101'],
            'date': ['20200915'],
            'page': ['2']}
    para['sid2'] = category
```

네이버 뉴스 경제 부문에서 하루에 있는 올라온 모든 페이지를 가져오고, 일주일 동안 올라온 뉴스 기사를 크롤링합니다.

```
# 일주일 동안 데이터 수집
for day in range(6,13):
    para['date'] = str(date(2020,9,day)).replace("-", "")
    para['page'] = 1000
    r = requests.get("https://news.naver.com/main/list.nhn?", params=para)
    # print (r.url)
    bs = BeautifulSoup(r.text)
    # 마지막 페이지 번호를 가져온다.
    b = bs.find('div', id='main_content')
    last_page = bs.find('div', class_='paging').find("strong").text
    #print("==========")
    #print (last_page)
    #print("==========")
    for page in range(1, int(last_page)+1):
        #print (page)
        para['page'] = page
        r = requests.get("https://news.naver.com/main/list.nhn?", params=para)
        bs = BeautifulSoup(r.text)

        for x in bs.find({"div" : "list_body newsflash_body"}).findAll("dl", class_=""):
            total = total.append(pd.DataFrame([[x.select("a")[-1].text.strip(), x.select("a")[-1]['href'],
                                                x.find({"span" : "lede"}).text, x.find("span", class_="writing").text,
                                                str(date(2020,9,day))]],
                                              columns=['title', 'url', 'content', 'company', 'date']))
return total
```

Komoran 을 이용한 한국어 형태소 분석

- 크롤링 해온 데이터에서 형태소 분석을 실행합니다. (뉴스 제목에서 형태소 분석을 실행)

```
df.iloc[0,0]
```

```
'[기하영의 생활 속 카드]해외여행 대신 해외직구…"카드 할인 챙기세요"'
```

```
tokenizer.pos(df.iloc[0,0])
```

```
[('[', 'SS'),
 ('기', 'NNG'),
 ('하영', 'NNP'),
 ('의', 'JKG'),
 ('생활', 'NNG'),
 ('속', 'NNG'),
 ('카드', 'NNG'),
 (']', 'SS'),
 ('해외여행', 'NNP'),
 ('대신', 'NNG'),
 ('해외직구', 'NNP'),
 ('…', 'SE'),
 ('"', 'SS'),
 ('카드', 'NNP'),
 ('할인', 'NNG'),
 ('챙기', 'VV'),
 ('시', 'EP'),
 ('어요', 'EC'),
 ('"', 'SS')]
```

형태소 분석 결과

- 명사일 경우 키워드로 채택한다.

NNG, NNP, NNB, NNM

일반 명사, 고유 명사,
수사와 단위 의존 명사 일 때
키워드로 채택합니다.

대분류	세종 품사 태그		심광섭 품사 태그		KKMA 단일 태그 V 1.0					
	태그	설명	Class	설명	묶음1	묶음2	태그	설명	확률태그	저장사전
체언	NNG	일반 명사	NN	명사	N	NN	NNG	보통 명사	NNA	noun.dic
	NNP	고유 명사					NNP	고유 명사		
	NNB	의존 명사	NX	의존 명사			NNB	일반 의존 명사	NNB	simple.dic
			UM	단위 명사			NNM	단위 의존 명사		
	NR	수사	NU	수사		NR	NR	수사	NR	
	NP	대명사	NP	대명사		NP	NP	대명사	NP	
용언	VV	동사	VV	동사	V	VV	VV	동사	VV	verb.dic
	VA	형용사	AJ	형용사		VA	VA	형용사	VA	
	VX	보조 용언	VX	보조 동사		VX	VXV	보조 동사	VX	
			AX	보조 형용사			VXA	보조 형용사		
	VCP	긍정 지정사	CP	서술격 조사 '이다'		VC	VCP	긍정 지정사, 서술격 조사 '이다'	VCP	raw.dic
	VCN	부정 지정사					VCN	부정 지정사, 형용사 '아니다'	VCN	

Komoran 을 이용한 한국어 형태소 분석

NNG, NNP, NNB, NNM 일 때 키워드를 만들도록 설정하고 Word Count를 실행합니다.
실행 후에는 조사, 부사와 같은 품사가 빠진 적합한 키워드를 생성 할 수 있습니다.

```
word_dict ={}
for cont in rt.title:
    for word, morpheme in token.pos(cont):
        if morpheme in ['NNG', 'NNP', 'NNB', "NNM"] and len(word) > 1 : #NNG NNP NNB NNM
            if word_dict.get(word) == None:
                word_dict[word] = 1
            else:
                word_dict[word] += 1
```

Komoran 을 이용한 한국어 형태소 분석

Stop word 지정 : 주목하고 싶은 특정 키워드가 있다면 stop word를 지정해 키워드를 구성할 수 있습니다. 금융, 대출에 stop word를 두고 진행한 결과입니다.

```
In [93]: stop_words = {'금융', '대출'}

In [94]: stop_words
Out[94]: {'금융', '대출'}

In [92]: '금융' in stop_words
Out[92]: True

In [96]: list(set([1,1,1,1,1,2,2,3,3,2,3,3,2,3]))
Out[96]: [1, 2, 3]

In [95]: word_dict2 ={}
         stop_words = {'금융', '대출'}
         for cont in rt2.title:
             for word, morpheme in token.pos(cont):
                 if morpheme in ['NNG', 'NNP', 'NNB', "NNM"] and len(word) > 1 and ₩
                     word not in stop_words:
                     if word_dict2.get(word) == None:
                         word_dict2[word] = 1
                     else:
                         word_dict2[word] += 1
```

Komoran 을 이용한 한국어 형태소 분석

Word Cloud를 통해 시각화를 진행합니다.

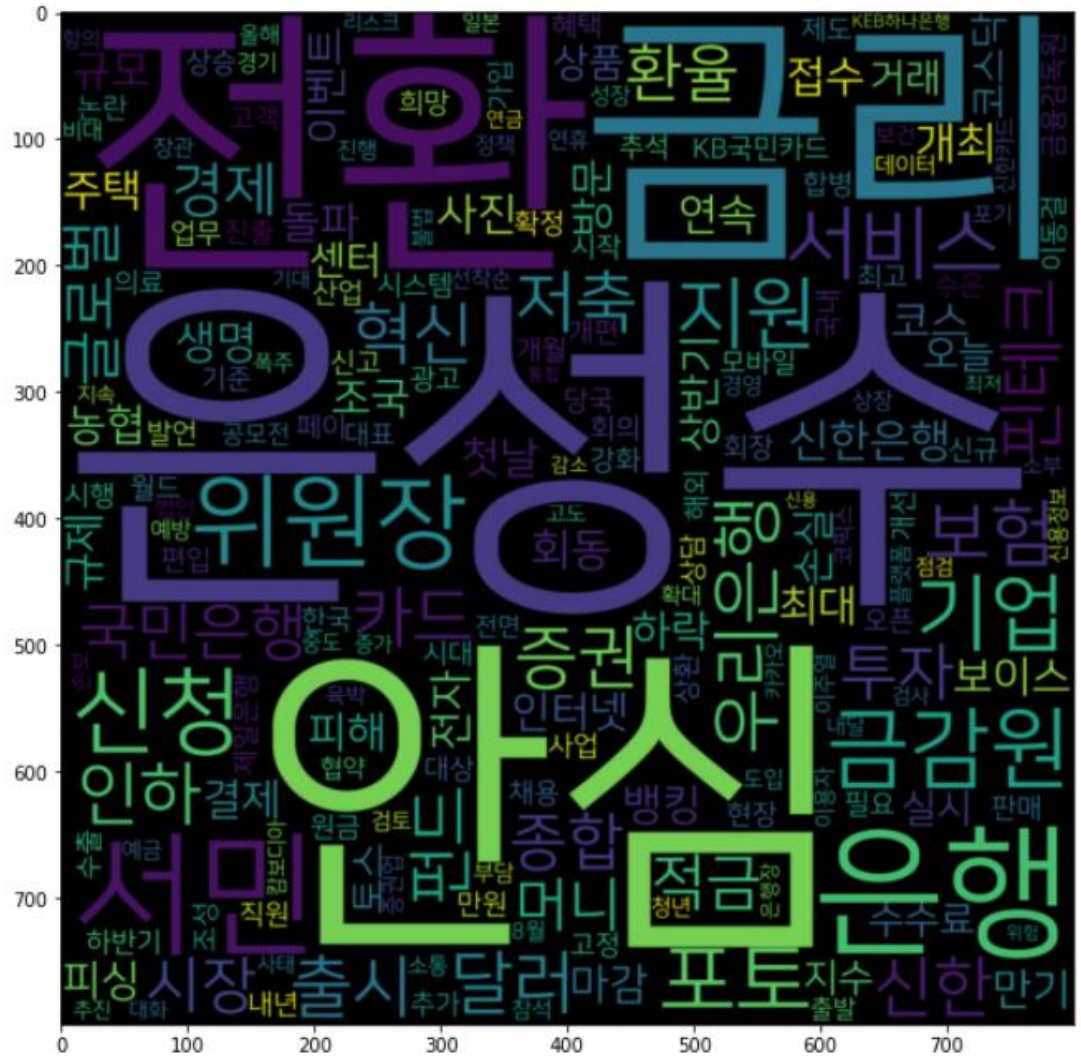
```
from wordcloud import WordCloud

wordcloud = WordCloud(
    font_path = 'C:/Windows/Fonts/나눔스퀘어OTF/NanumSquareR.otf',
     width = 800,
     height = 800
)

wordcloud = wordcloud.generate_from_frequencies(keywords)
```

```
array = wordcloud.to_array()
print(type(array)) # numpy.ndarray
print(array.shape) # (800, 800, 3)
```

```
<class 'numpy.ndarray'>
(800, 800, 3)
```



총평

이번 기회를 통해 Komoran을 알 수 있었고, 이를 이용해 NLP의 기초를 닦을 수 있었습니다.

크롤링한 뉴스 제목을 가지고 각각의 형태소로 분해에 적시적소에 맞게 사용할 수 있다는 점이 흥미로웠습니다.

기회가 된다면 NLP 기술을 이용해 사용자의 감정을 분석하는 서비스 플랫폼을 개발하고 싶습니다.