댓글 데이터를 통한 뉴스 여론 분석

7조_이민서, 이소연, 정승혜, 황지원

목차

1.
주제 선정 배경

01. 주제 선정 배경

- ▷ 네이버의 실시간 검색어 서비스 종료(2021.02)
- ▷ 신문자, 기자 성향에 따라 편향 多
- ▷ 뉴스 내용과 여론이 과연 얼마나 일치할까?

2.
분석 개요

02. 분석 개요

분석 채널

NAVER

분석 도구

pandas

NumPy

Selenium

Beautiful Soup

matplotlib

분석 설명

- 수집 기간 : 2021.11.10 ~ 2021.11.30
- 각 당의 대선 후보 선출 확정 이후 11월10일부터 11월 30일까지의 데이터를 수집

3.
프로젝트 과정

(1)데이터 선택

1. 웹 크롤링
- beautifulsoup4, selenium

(1)데이터 선택

2. Twitter Crawling
   - Tweepy
   - Twint

```
import twint


def grab_tweets(search, file, since):
    c = twint.Config()
    c.Search = search
    c.Since = since
    c.Hide_output = True
    c.Store_json = True
    c.Output = file
    twint.run.Search(c)
```

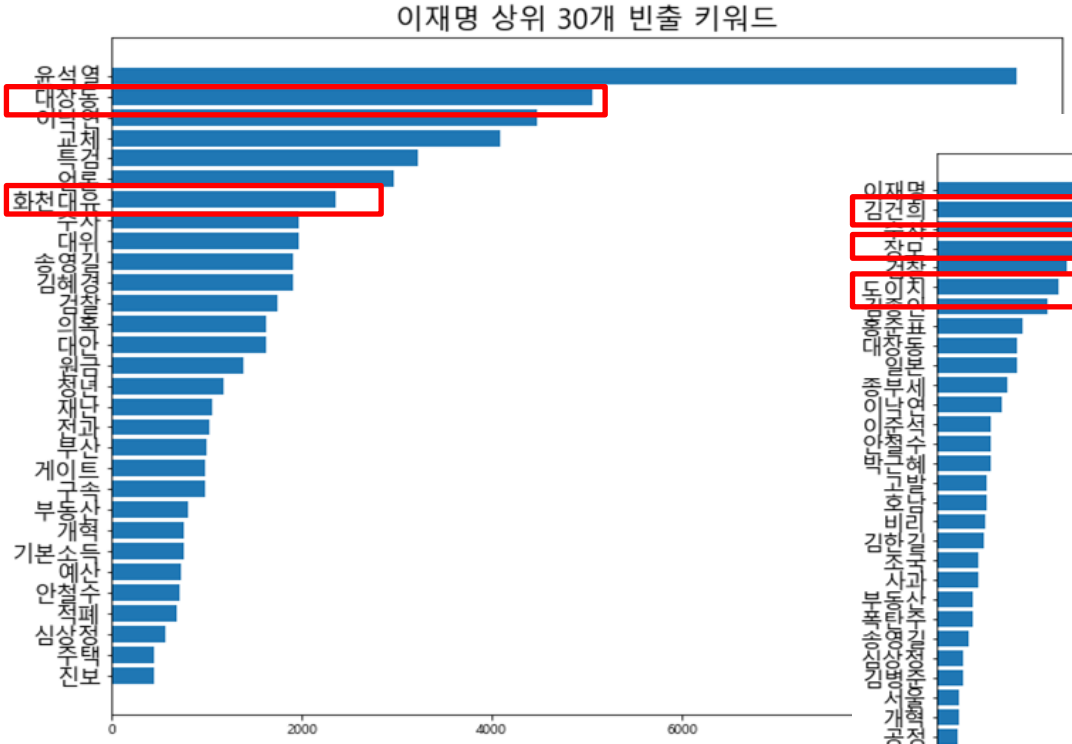
(1)데이터 선택

3. 데이터 전처리
   - 한글의 자음, 모음 제외(ㅋㅋ, ㅎㅎ, ㅡㅡ 등)
   - 띄어쓰기 및 줄 바꿈 제거
   - 특수 기호 및 이모티콘 제거
   - 맞춤법 검사
     - Hanspell
     - 부산대 맞춤법 교정기
       (http://speller.cs.pusan.ac.kr/)

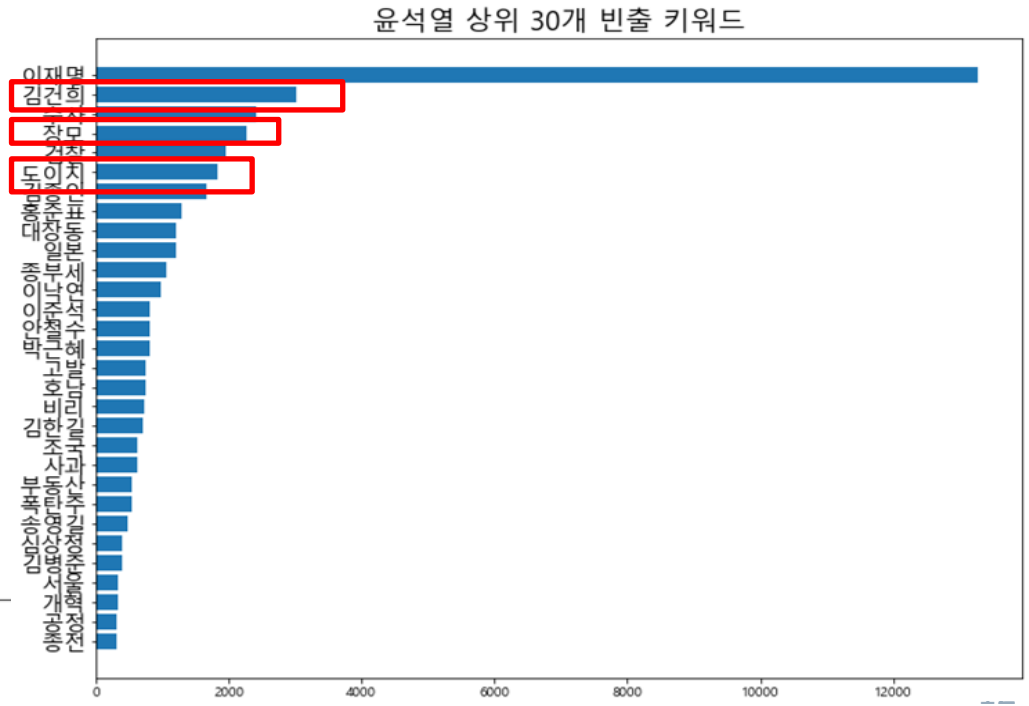
(2) 키워드 분석 -형태소 분석

```
from konlpy.tag import Okt
from collections import Counter
from ckonlpy.tag import Twitter
print("\n준비과정 4 - 명사 단위, 형태소 단위 등등으로 쪼개기")
# 형태소 추출도구로 konlpy의 Okt와 Twitter 사용해보기
# ckonlpy는 Customized Konlpy로 Konlpy의 customized version이라고 보면 된다.
# ++ 추가 UserWarning 메시지 내용 : KoNLPy 0.4.5 버전부터는 Twitter 패키지 이름이 Okt로 바꼈다. -> 에러는 안남. 편의상 코드에서는 Twitter로 표현하겠음!

print("\n명사 단위로 쪼개기")
# 명사 단위로 쪼개서 nouns_txt에 담기
# ++ 추가 ex) '절대 반대'를 '절대' + '반대'로 쪼개서 담는게 싫다면 Twitter 이용해서 임의로 명사 추가하기!('절대 반대'가 한 단어로 취급)

Twitter = Twitter()
Twitter.add_dictionary('아프간난민', 'Noun')
Twitter.add_dictionary('아프간 난민', 'Noun')

# Twitter의 add_dictionary()를 통해서 명사 추가하고, Okt를 통해 다시 한번 명사 쪼개면 해결!
nouns_txt = okt.nouns(content_all)
print(nouns_txt)

print("\n형태소 단위로 쪼개기")
morphs_txt = okt.morphs(content_all)
print(morphs_txt)
```

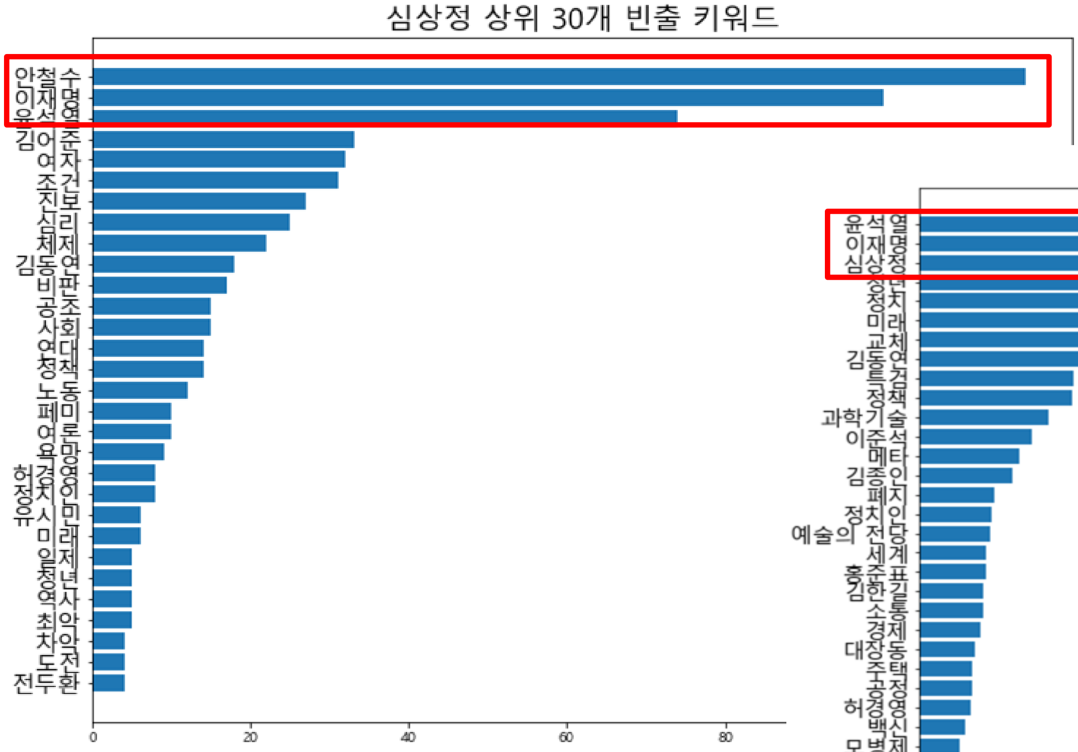
(2) 키워드 분석 - 빈도 분석



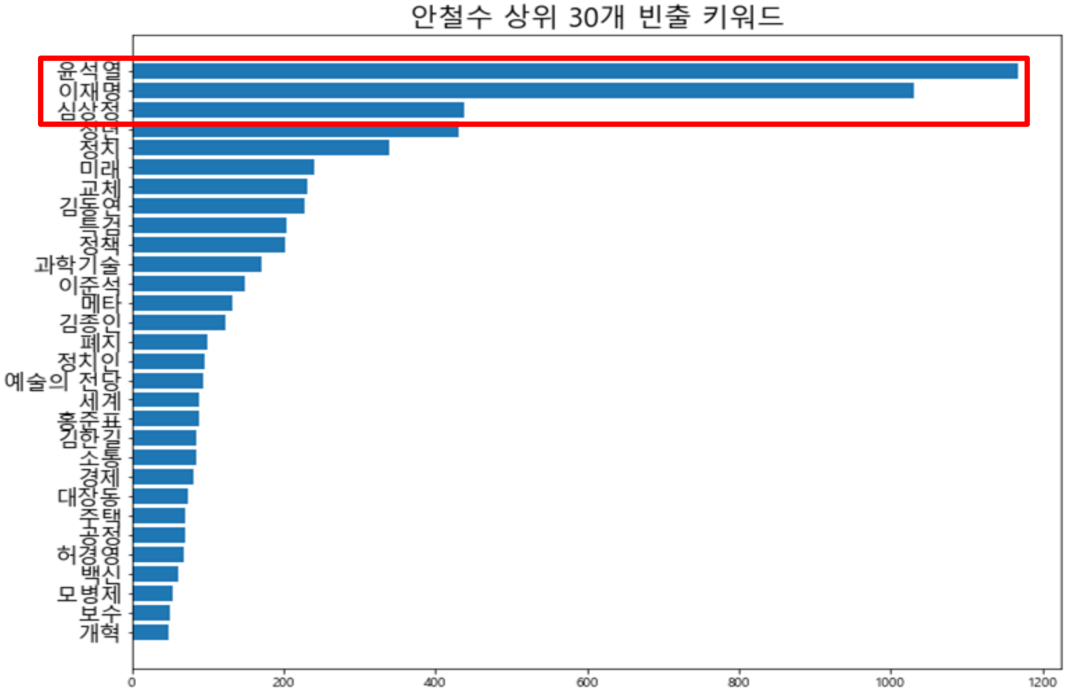
수집 기간 : 2021년 11월 10일 ~ 11월 30일



(2) 키워드 분석 - 빈도 분석



심상정 상위 30개 빈출 키워드
안철수
이재명
윤석열
김어준
여자
조건
진보
심리
체제
김동연
비판
공조
사회
여대
정책
노동
페미
여론
욕망
허경영
정치인
유시민
미래
일제
청년
역사
최악
차악
도전
전두환
0
20
40
60
80



안철수 상위 30개 빈출 키워드
윤석열
이재명
심상정
청년
정치
미래
교체
김동연
특검
정책
과학기술
이준석
메타
김종인
폐지
정치인
예술의 전당
세계
홍준표
김한길
소통
경제
대장동
주택
공정
허경영
백신
모병제
보수
개혁
0
200
400
600
800
1000
1200

(2) 키워드 분석 - 워드 클라우드

In [12]:
```
from konlpy.tag import Okt

okt = Okt()

def tokenize(doc):
    return [t for t in okt.pos(doc, norm=True, stem=True)]      # norm은 정규화, stem은

tokenize_docs = [(tokenize(row[0]), row[1]) for row in data_split]

with open('/content/drive/MyDrive/Colab Notebooks/tokenized_data', 'w', encoding="utf-8"
    json.dump(tokenize_docs, make_file, ensure_ascii=False, indent="\t")
tokenize_docs[0]
```

Out[12]:
```
([('"', 'Punctuation'),
  ('윤석열', 'Noun'),
  ('이재명', 'Noun'),
  ('인성', 'Noun'),
  ('논란', 'Noun'),
  ('이', 'Noun'),
  ('영상', 'Noun'),
  ('하나로', 'Noun'),
  ('종결', 'Noun'),
  ('짓다', 'Verb'),
  ('심각하다', 'Adjective'),
  ('언론', 'Noun'),
  ('형평성', 'Noun'),
  ("'", 'Punctuation'),
  (',', 'Punctuation'),
  ('"', 'Punctuation'),
  ('좌파', 'Noun'),
  ('를', 'Suffix'),
  ('이', 'Josa'),
  ('이재명', 'Noun'),
  ('을', 'Josa'),
  ('구속', 'Noun'),
```

In [14]:
```
# 명사만 추출
noun_count = [x[0][0]:x[1] for x in count_words if x[0][1]=='Noun' and len(x[0][0])>=2
noun_count
```

Out[14]:
```
{'이재명': 73195,
 '후보': 15638,
 '민주당': 13198,
 '윤석열': 9512,
 '뉴스': 7569,
 '국민': 6613,
 '대통령': 5806,
 '대장동': 5060,
 '다음': 5030,
 '이낙연': 4475,
 '사람': 4393,
 '대선': 4384,
 '교체': 4078,
 '특검': 3225,
 '지지율': 3130,
 '지금': 3064,
 '출처': 3064,
 '언론': 2965,
 '생각': 2766,
 '지지': 2649,
 '문재인': 2482,
 '우리': 2276,
 '지금': 2094,
```

(3) 키워드 분석 - TF-IDF

- 벡터화

```
vectorizer = CountVectorizer(analyzer = 'word', # 캐릭터 단위로 벡터화 할 수도 있습니다.
                             tokenizer = None, # 토크나이저를 따로 지정해 줄 수도 있습니다.
                             preprocessor = None, # 전처리 도구
                             stop_words = None, # 불용어 nltk등의 도구를 사용할 수도 있습니다.
                             min_df = 2, # 토큰이 나타날 최소 문서 개수로 오타나 자주 나오지 않는 특수한 전문용어 제거에 좋습니다.

                            )
```

- 단어 벡터를 더함

```
dist = np.sum(feature_vector, axis=0)

df_freq = pd.DataFrame(dist, columns=vocab)
df_freq
```

	00	000	000 대	000 명	000 원	0011_9561	00 때문에	01	010	0189	...	힘이라도	힘이지만	힘입니다	힘입었다	힘있는	힘주고
0	27	6	2	4	3	3	3	18	5	26	...	3	2	2	4	2	2

1 rows × 65880 columns

(3) 키워드 분석 - TF-IDF

- 빈도수로 정렬

```
df_freq.T.sort_values(by=0, ascending=False).head(30)
```

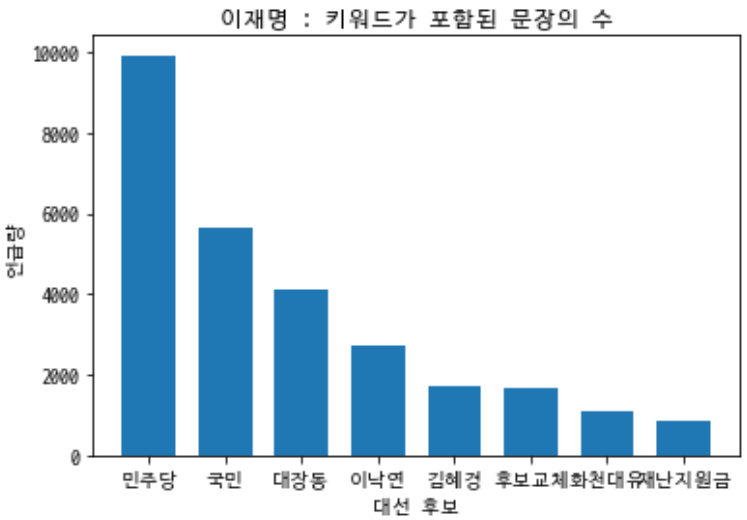
- TF-IDF로 가중치를 주어 벡터화

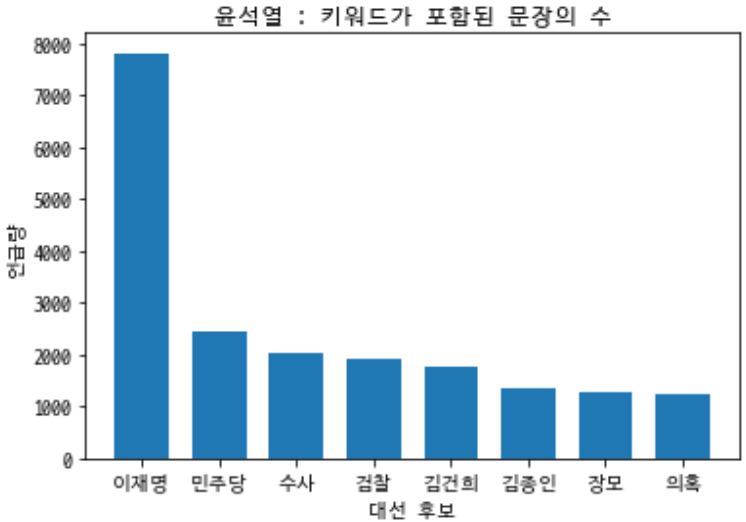
```
from sklearn.feature_extraction.text import TfidfTransformer
transformer = TfidfTransformer(smooth_idf=False)
transformer
```

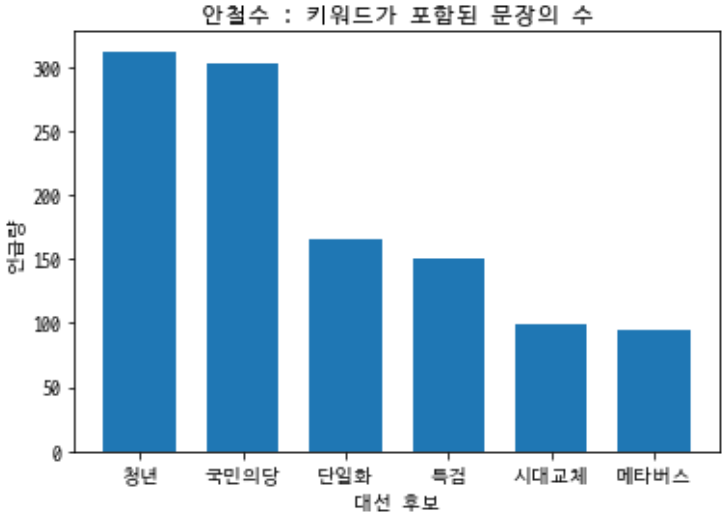
```
feature_tfidf = transformer.fit_transform(feature_vector)
```

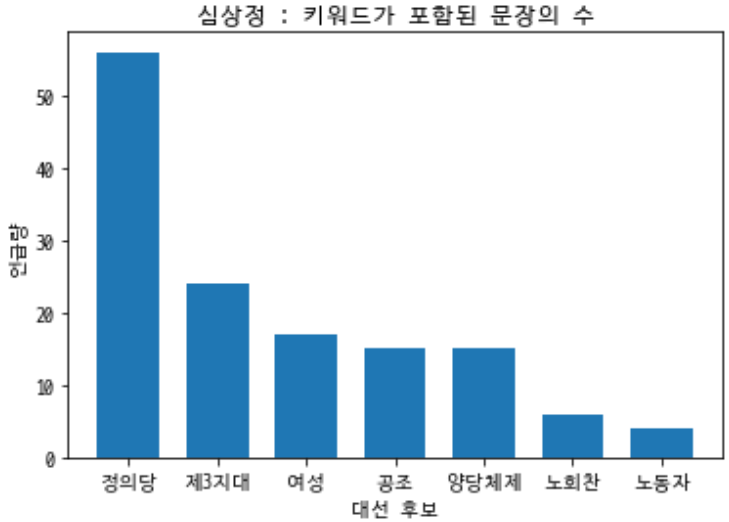
- 가중치 더함

```
tfidf_freq = pd.DataFrame(feature_tfidf.toarray(), columns=vocab)
```

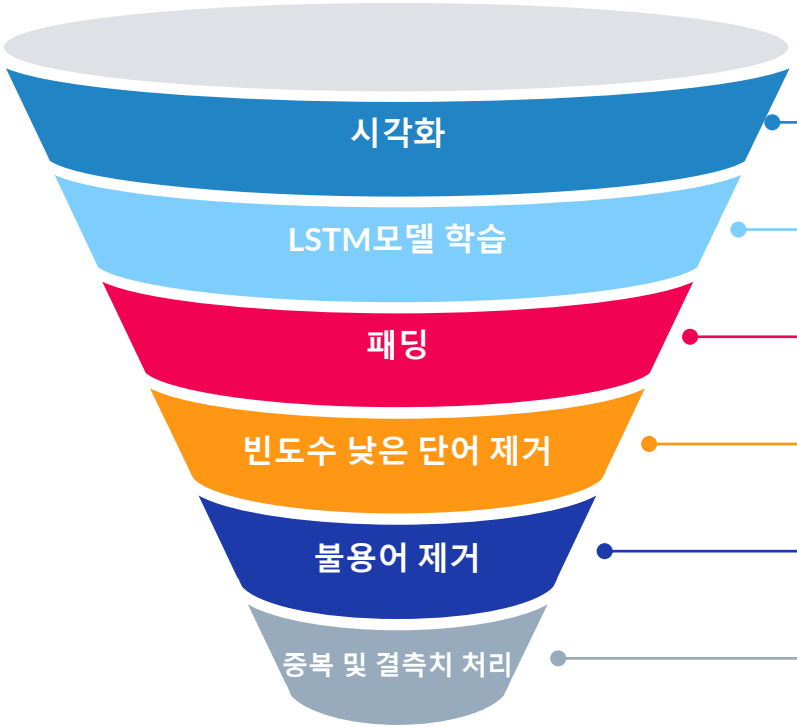

이재명 : 키워드가 포함된 문장의 수
언급량
대선 후보
민주당
국민
대장동
이낙연
김혜경
후보교체
화천대유
재난지원금


윤석열 : 키워드가 포함된 문장의 수
언급량
대선 후보
이재명
민주당
수사
검찰
김건희
김종인
장모
의혹


안철수 : 키워드가 포함된 문장의 수
언급량
대선 후보
청년
국민의당
단일화
특검
시대교체
메타버스


심상정 : 키워드가 포함된 문장의 수
언급량
대선 후보
정의당
제3지대
여성
공조
양당체제
노회찬
노동자

(4) 감정분석



- 도넛차트 시각화를 통해 긍정 부정 비교
- 감정상태분류모델 LSTM사용
- 패딩단계를 통한 전반적인 길이 조정
- 빈도수 낮은 단어 학습에 영향에 도움되지 않음
- 불용어 사전(한국어 코퍼스)을 이용하여 단어를 분리하고 불용어 제거
- 중복값과 결측값 제외

(4) 감정분석

기계학습을 이용한 감정 분석

한국어 자연어 처리 konlpy와 형태소 분석기 MeCab 설치

- https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh

```
[ ]  1 #!set -x
     2 ! pip install konlpy
     3 ! curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh | bash -x
```

```
Collecting konlpy
  Downloading konlpy-0.5.2-py2.py3-none-any.whl (19.4 MB)
```

▾ 토큰화 및 불용어 제거

- 단어들을 분리하고 불용어를 제거함
- 불용어 사전: '의','가','이','은','를','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다'

```
[ ]  1 stopwords=['의','가','이','은','를','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다']
```

```
[ ]  1 mecab = Mecab()
     2
     3 X_train = []
     4 for sentence in train_data['document']:
     5   X_train.append([word for word in mecab.morphs(sentence) if not word in stopwords])
```

```
[ ]  1 print(X_train[:3])
```

```
[ ]  1 X_test = []
     2 for sentence in test_data['document']:
     3   X_test.append([word for word in mecab.morphs(sentence) if not word in stopwords])
```

```
[ ]  1 tokenizer = Tokenizer()
     2 tokenizer.fit_on_texts(X_train)
     3 print(tokenizer.word_index)
```

▾ 빈도 수가 낮은 단어 제거

▾ 중복 및 결측치 처리

- 데이터 개수 확인
- 데이터에 중복이 존재한다면 이를 제거

```
[ ]  1 print(train_data['document'].nunique())
     2 print(train_data['label'].nunique())
     3
     4 train_data.drop_duplicates(subset=['document'], inplace=True)
```

```
[ ]  1 print(train_data.isnull().sum())
     2
     3 train_data = train_data.dropna(how='any')
```

▾ 패딩

- 리뷰의 전반적인 길이를 확인
- 모델의 입력을 위해 동일한 길이로 맞춰줌

```
[ ]  1 print('리뷰 최대 길이:', max(len(l) for l in X_train))
     2 print('리뷰 평균 길이:', sum(map(len, X_train))/len(X_train))
```

```
[ ]  1 plt.hist([len(s) for s in X_train], bins=50)
     2 plt.xlabel('Length of Samples')
     3 plt.ylabel('Number of Samples')
     4 plt.show()
```

```
[ ]  1 max_len = 60
```

```
[ ]  1 X_train = pad_sequences(X_train, maxlen=max_len)
     2 X_test = pad_sequences(X_test, maxlen=max_len)
```

▾ 모델 구축 및 학습

- 감정 상태 분류 모델을 선언하고 학습
- 모델은 일반적인 LSTM 모델을 사용

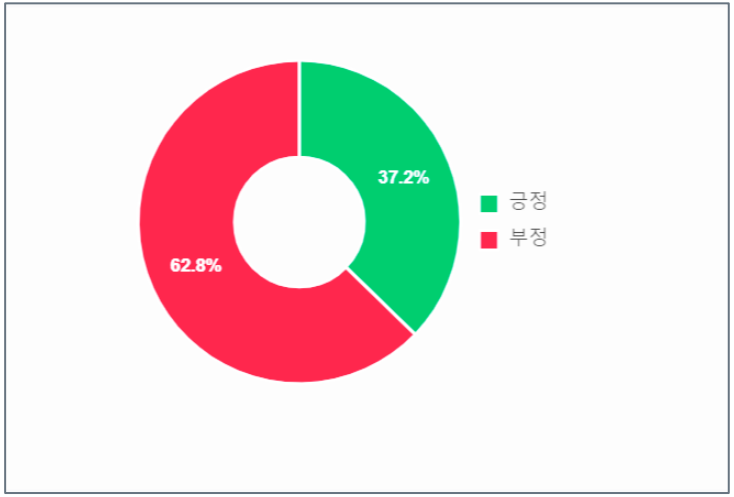
```
[ ]  1 from tensorflow.keras.layers import Embedding, Dense, LSTM
     2 from tensorflow.keras.models import Sequential
```

```
[ ]  1 model = Sequential()
     2 model.add(Embedding(vocab_size, 100))
     3 model.add(LSTM(128))
     4 model.add(Dense(1, activation='sigmoid')) #오버피팅 없애려면 dropout 해줘!
     5
     6 model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
     7 model.summary()
```

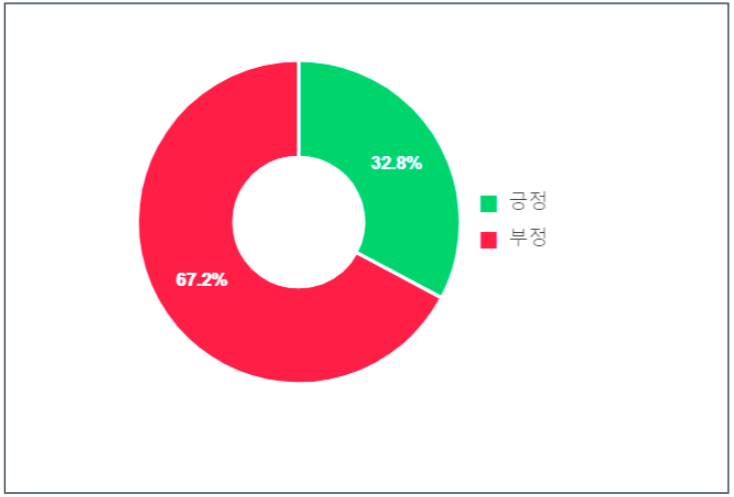
```
[ ]  1 history = model.fit(X_train, y_train, epochs=15, batch_size=60, validation_split=0.2)
```

```
[ ]  1 model.evaluate(X_test, y_test)
```

(4) 감정분석



이재명 후보에 대한 댓글의 긍부정 분포비율



윤석열 후보에 대한 댓글의 긍부정 분포비율

(4) 감정분석

이재명 후보에 대한 댓글의 긍부정 키워드 분포

윤석열 후보에 대한 댓글의 긍부정 키워드 분포

(5) 토픽모델링

형태소분석 및 토큰화	정수인코딩	LDA모델 학습	주어진 키워드에 관련된 단어찾기
• 한글 자연어 처리 kkma 이용	• Gensim의 copora Dictionary 이용 • 자주 언급되는 키워드 찾기 • word_id와 word_frequency에 저장	• 20개의 토픽을 추출하여 각 토픽별 기여하는 키워드	• 각 후보별 정책의 키워드를 넣어 함께 나오는 단어 탐색

(5) 토픽모델링

- 정수 인코딩

정수 인코딩과 단어 집합 만들기

```
!pip install konlpy
from konlpy.tag import Kkma
kkma = Kkma()
tokenized_doc=[]
for n in range(len(data["tweet"])):
    text = data["tweet"][n]
    tokenized_doc.append(list(kkma.nouns(str(text))))
print(tokenized_doc)
```

```
Requirement already satisfied: konlpy in /usr/local/lib/python3.7/dist-packages (0.5.2)
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.7/dist-packages (fr
```

- 모델 학습

```
import gensim
NUM_TOPICS = 20 #20개의 토픽(카테고리), k=20
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictionary, passes=15)
topics = ldamodel.print_topics(num_words=7)
for topic in topics:
    print(topic)
```

```
(0, '0.030*"잘모" + 0.025*"윤" + 0.020*"가족" + 0.015*"평" + 0.015*"억" + 0.012*"병원" + 0.011*"요양"')
(1, '0.040*"윤" + 0.033*"대결" + 0.023*"2" + 0.019*"문" + 0.018*"뉴스" + 0.015*"다음" + 0.015*"침묵"')
(2, '0.039*"윤" + 0.020*"50" + 0.020*"부산" + 0.017*"수사" + 0.017*"열" + 0.017*"특검" + 0.015*"석"')
(3, '0.028*"윤" + 0.020*"길" + 0.016*"송영" + 0.015*"조선" + 0.014*"송영길" + 0.014*"면제" + 0.013*"일보"')
(4, '0.031*"윤" + 0.027*"대한" + 0.027*"민국" + 0.027*"대한민국" + 0.019*"대통령" + 0.017*"열" + 0.014*"검찰"')
(5, '0.027*"윤" + 0.025*"총장" + 0.022*"검찰" + 0.019*"검찰총장" + 0.016*"열" + 0.011*"오" + 0.010*"조국"')
(6, '0.056*"윤" + 0.050*"열" + 0.053*"이재명" + 0.039*"뉴스" + 0.023*"1" + 0.023*"출처" + 0.022*"4"')
(7, '0.053*"후보" + 0.041*"윤" + 0.034*"대선" + 0.034*"힘" + 0.029*"국민" + 0.028*"열" + 0.019*"이재명"')
(8, '0.049*"표" + 0.048*"돈" + 0.044*"윤" + 0.043*"대장" + 0.040*"홍" + 0.040*"대장동" + 0.030*"국민"')
(9, '0.053*"김" + 0.048*"윤" + 0.035*"위" + 0.032*"선대" + 0.025*"선대위" + 0.023*"열" + 0.023*"증인"')
(10, '0.048*"윤" + 0.044*"종" + 0.040*"부세" + 0.039*"종부세" + 0.020*"열" + 0.020*"1" + 0.014*"부자"')
(11, '0.025*"윤" + 0.022*"기" + 0.021*"례" + 0.017*"기례" + 0.015*"열" + 0.014*"폭탄" + 0.014*"재검토"')
(12, '0.037*"문" + 0.032*"재인" + 0.031*"윤" + 0.030*"수사" + 0.030*"처" + 0.026*"문재인" + 0.026*"공수"')
(13, '0.079*"윤" + 0.059*"열" + 0.038*"이재명" + 0.024*"거" + 0.019*"대통령" + 0.015*"후보" + 0.015*"사람"')
(14, '0.043*"윤" + 0.043*"김" + 0.039*"김건희" + 0.032*"건희" + 0.023*"조작" + 0.022*"열" + 0.020*"주가"')
(15, '0.026*"윤" + 0.023*"박근" + 0.022*"혜" + 0.022*"박근혜" + 0.017*"이명" + 0.015*"월" + 0.014*"박"')
(16, '0.041*"윤" + 0.030*"석" + 0.016*"박" + 0.016*"열도" + 0.016*"윤석열도" + 0.016*"열" + 0.015*"수"')
(17, '0.022*"윤" + 0.013*"김" + 0.013*"경" + 0.013*"열" + 0.010*"김혜경" + 0.010*"혜" + 0.009*"관계"')
(18, '0.018*"일본" + 0.018*"윤" + 0.017*"정권" + 0.016*"반대" + 0.015*"선언" + 0.015*"호남" + 0.012*"종전"')
(19, '0.056*"윤" + 0.041*"열" + 0.018*"뉴스" + 0.017*"사과" + 0.017*"다음" + 0.013*"5" + 0.012*"1"')
```

- 문서 별 토픽 중요도

```
topictable = make_topictable_per_doc(ldamodel, corpus)
topictable = topictable.reset_index() # 문서 번호을 의미하는 열(column)로 사용하기 위해서 인덱스 열을 하나 더 만든다
topictable.columns = ['문서 번호', '가장 비중이 높은 토픽', '가장 높은 토픽의 비중', '각 토픽의 비중']
topictable[:10]#상위 10개 댓글별 비중이 높은 토픽은 무엇이고 얼만큼 차지하는지
```

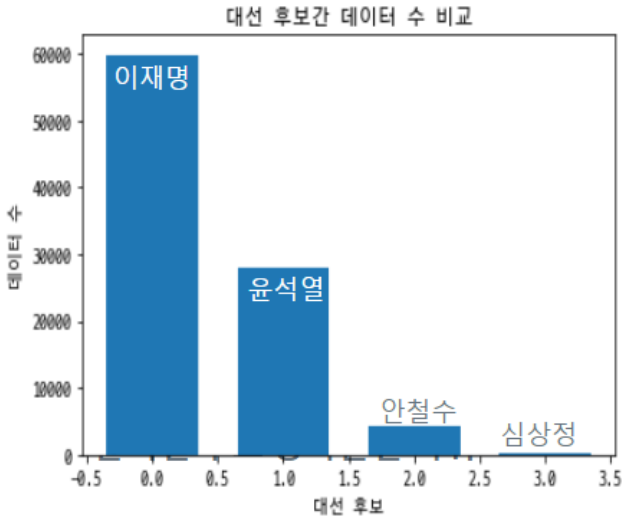
	문서 번호	가장 비중이 높은 토픽	가장 높은 토픽의 비중	각 토픽의 비중
0	0	10.0	0.3264	[(0, 0.1553484), (4, 0.06819149), (6, 0.075518...
1	1	19.0	0.4453	[(3, 0.20335414), (8, 0.20972669), (19, 0.4452...
2	2	13.0	0.8944	[(13, 0.89444447)]
3	3	18.0	0.4569	[(9, 0.072563685), (11, 0.07815084), (12, 0.07...
4	4	13.0	0.5036	[(1, 0.17883931), (5, 0.039845828), (13, 0.503...
5	5	1.0	0.4835	[(1, 0.48346618), (7, 0.16024579), (9, 0.03316...
6	6	1.0	0.6510	[(1, 0.6510349), (6, 0.22995703), (13, 0.08631...
7	7	1.0	0.6309	[(1, 0.63089097), (2, 0.20014979), (3, 0.06229...
8	8	0.0	0.4686	[(0, 0.46860486), (6, 0.09392841), (9, 0.29396...
9	9	18.0	0.9721	[(18, 0.9720588)]

- 윤석열 후보의 세금 관련 키워드

```
19, '0.056*"윤" + 0.041*"열" + 0.018*"뉴스" + 0.017*"사과" + 0.017*"다음" + 0.013*"5" + 0.012*"1"')
10, '0.048*"윤" + 0.044*"종" + 0.040*"부세" + 0.039*"종부세" + 0.020*"열" + 0.020*"1" + 0.014*"부자"')
4, '0.031*"윤" + 0.027*"대한" + 0.027*"민국" + 0.027*"대한민국" + 0.019*"대통령" + 0.017*"열" + 0.014*"검찰"')
```

(5) 토픽모델링

- 대선 후보간 데이터 수 비교



- 이재명 후보 공약관련 키워드

	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
	기본소득 지급	탄소세	52시간 주 4일제 도입, 근로시간 단축	코로나 대응 – 전국민 재난지원금(철회, 재검토)	공공의료	선택적 모병제
1	이재명	환경	노동	철회	확대	군대
2	사람	기후	경제	국민	국민	북한
3	지지	발전	청년	대장동	코로나	전쟁
4	생각	신재생	취업	특검	접근성	의무
5	뉴스	위기	뉴스	예산	체계	문제

- 윤석열 후보 공약관련 키워드

	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
	신규주택 공급	세금 대폭 인하	자율적 추가근로	재난지원금 반대	모병제 반대	군필자에 대한 민간주택 청약 가산점 제도 도입
1	종부세	종부세	근로세	이재명	전쟁	군대
2	대출	정치	철회	철회	북한	전쟁
3	청년	부자	일자리	세금	군대	복무
4	공약	부동산	안철수	혈세	종전	미필
5	부자	의심	종부세	부자	병무청	청약통장

4.
개선 사항

4. 개선 사항

- 대선 후보별로 동의어 사전 만들기
- 대선 토론으로 알아보는 대선 주자별로 어떤 정책에 어떤 스탠스를 취하는지
- 대선 주자별 현대통령과의 정책 및 스탠스 비교

5.
프로젝트 정리

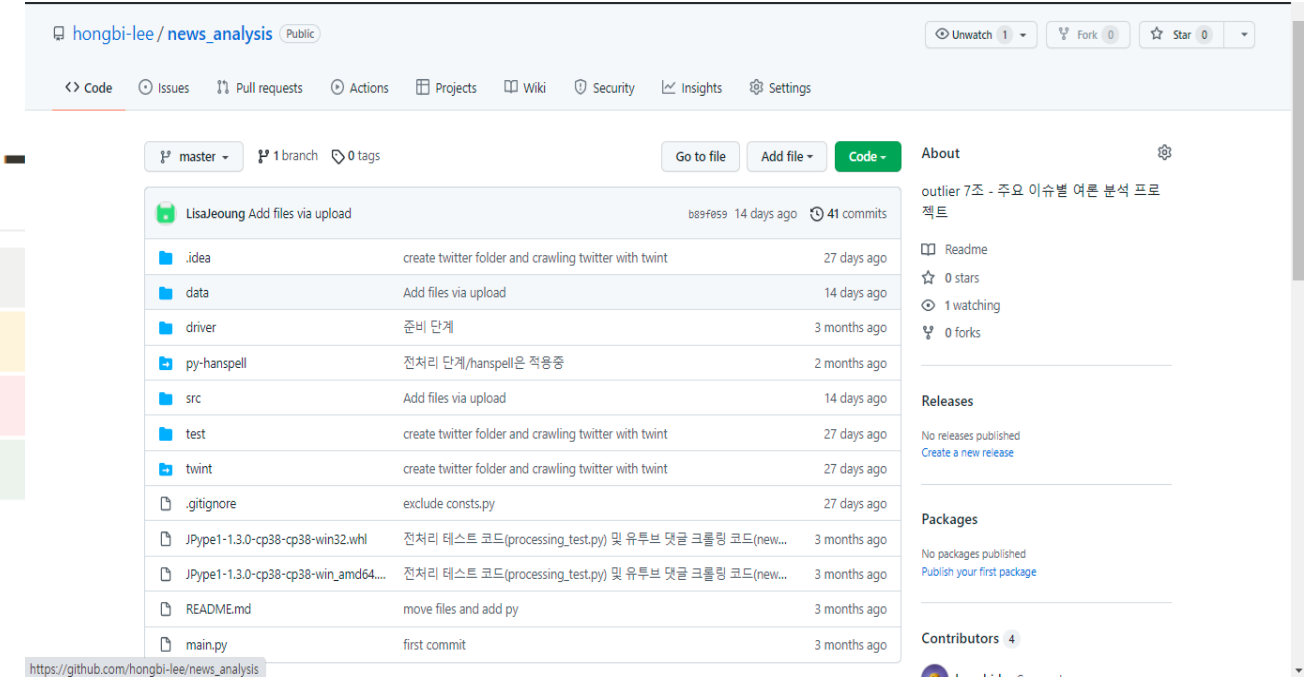
5. 프로젝트 정리

- https://github.com/hongbi-lee/news_analysis

커버 추가 댓글 추가

OUTLIER 7조 -

- 개요
- Projects Management
- 작업 목록
- 회의록
- Github
- 구글 드라이브



감사합니다