

빅데이터 최신기술 최종 과제 보고서

컴퓨터공학부

20163084 권보경

- * 사용 언어 : Python 3.7
- * 사용 텍스트 : KCCq28_Q01.txt (45.3 MB, 20만 문장, 약 412만 어절)
- * unigram 개수 635598 / 10개 이상 : 5603개 / 3개 이상 : 239423 개
- * bigram 개수 : 2975633 / 10개 이상 : 6981개 / 3개 이상 : 4095164 개
- * trigram 개수 : 3943837 / 10개 이상 : 8525개 / 3개 이상 : 1200534 개

1. 어절 단위 빈도 조사

- 텍스트에서 문장의 시작과 끝을 표시하는 코드 (시작 : SS , 끝 : SE)

```
import operator

def main():
    f= open('KCCq28_Korean_sentences_UTF8.txt', 'r') # 읽을 파일
    fw = open('KCCq28_Korean_sentences.txt','w') # 출력할 파일

    while True:
        line = f.readline()
        if not line: break
        fw.write(" SS " + line + " SE ") # 문장의 시작과 끝 표시

    f.close()
    fw.close()

if __name__ == '__main__':
    main()
```

- unigram 빈도 조사 전체 코드

```
import operator
import string
import ast

# ngram 처리하는 함수
def ngram(sentence, n):
    # 특수문자, 개행 제거
    sentence = sentence.replace('\n', ' ').replace('\r', ' ')
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)] # 리스트 형태로 텍스트 저장
```

```

return ngrams

# ngram 빈도 계산하는 함수, 딕셔너리 형태로 반환
def make_freqlist(ngrams):
    freqlist = {}

    for i in ngrams:
        if (i in freqlist): # 해당 word 가 있는 경우 빈도 count
            freqlist[i] += 1
        else:
            freqlist[i] = 1

    return freqlist

def uni_list(sorted_freqlist) :
    uni_list = []

    uni_list= sorted_freqlist[:3]

    return uni_list

def main():
    with open('KCCq28_Korean_sentences.txt', 'r', newline='\n') as f: # 텍스트 파일 열기
        file = f.read() # f에 담긴 텍스트 파일 읽어오기
        sample_file = ''.join(file) # 리스트를 문자열로 바꾼 뒤 변수에 저장

    fw = open('KCCtrigram.txt','w') # 출력할 파일

    # unigram을 처리하기 위해 텍스트 파일과 1을 매개변수로 함수 실행
    ngrams = tuple(ngram(file, 1))
    freqlist = make_freqlist(ngrams) # 빈도 수를 계산한 리스트를 freqlist에 저장

    # 빈도 수를 기준으로 정렬
    sorted_freqlist = sorted(freqlist.items(), key=operator.itemgetter(1), reverse=True).

    fw.write(str(dict(sorted_freqlist)))
    f.close()
    fw.close()

```

```

if __name__ == '__main__':
    main()

```

- unigram 빈도 조사 결과

{('말했다.',): 47099, ('수',): 36443, ('밝혔다.',): 20903, ('것"이라고',): 19274, ('있는',): 19153, ('그는',): 18469, ('관계자는',): 16661, ('있다"고',): 15484, ('한',): 13688, ('이',): 13520, ('등',): 13367, ('대한',): 13354, ('대해',): 12885, ('통해',): 12236, ('전',): 12131, ('있다"며',): 11247, ('위해',): 10692, ('이어',): 10245, ('설명했다.',): 10204, ('한다"고',): 9766, ('할',): 9622, ('강조했다.',): 9530, ('것으로',): 9376, ('이날',): 9135, ('것은',): 8964, ('김',): 8645, ('더',): 8602, ('또',): 8323, ('했다.',): 8242, ('것"이라며',): 7590, ('의원은',): 7209, ('것이',): 7145, ('박',): 7022, ('하는',): 6966, ('안',): 6958, ('덧붙였다.',): 6951, ('될',): 6611, ('전했다.',): 6498, ('것',): 6443, ('때문에',): 6428, ('위한',): 6226, ('대표는',): 6155, ('그',): 5959, ('있도록',): 5585, ('것을',): 5534, ('함께',): 5506, ('있다.',): 5493, ('개',): 5030, ('지적했다.',): 4974, ('없다"고',): 4821, ('및',): 4739, ('때',): 4724, ('많은',): 4724, ('우리',): 4719, ('큰',): 4688, ('잘',): 4540, ('그러면서',): 4522, ('등을',): 4486, ('주장했다.',): 4433, ('한다"며',): 4395, ('다른',): 4367, ('대통령은',): 4348, ('아니라',): 4258, ('이에',): 4151, ('이번',): 4095, ('하고',): 4074, ('없는',): 4070, ('지난',): 4062, ('있을',): 4015, ('같은',): 3959, ('많이',): 3766, ('감독은',): 3666, ('문',): 3552, ('모든',): 3539, ('관련',): 3491, ('가장',): 3467, ('경우',): 3466, ('이같이',): 3390, ('따라',): 3388, ('있어',): 3289, ('대해서는',): 3240, ('좋은',): 3207, ('새로운',): 3195, ('종',): 3143, ('이번',): 3143, ('만큼',): 3137, ('다양한',): 3085, ('없다"며',): 3080, ('대변인은',): 3070, ('다시',): 3036, ('데',): 3034, ('뒤',): 2941, ('교수는',): 2917, ('않고',): 2887 ('모드',): 2816 ('여러',): 2802 ('당시',): 2775 ('이후',): 2774 ('이력',): 2706

- bigram 빈도 조사 코드

```

# unigram 빈도 조사 코드와 거의 동일하며, 아래의 함수에서 매개변수 n의 값을 2로 주면 됨
def ngram(sentence, n):
    # 특수문자, 개행 제거
    sentence = sentence.replace('\n', ' ').replace('\r', ' ')
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)] # 리스트 형태로 텍스트 저장
    return ngrams

```

- bigram 빈도 조사 결과

{('SE', 'SS'): 1337720, ('말했다.', 'SE'): 318209, ('밝혔다.', 'SE'): 139514, ('SS', '그는'): 99278, ('설명했다.', 'SE'): 67874, ('강조했다.', 'SE'): 63785, ('했다.', 'SE'): 54592, ('수', '있는'): 51532, ('SS', '이어'): 51311, ('덧붙였다.', 'SE'): 44840, ('전했다.', 'SE'): 43330, ('것"이라고', '말했다.'): 43125, ('SS', '이'): 42172, ('수', '있도록'): 37224, ('있다.', 'SE'): 36908, ('SS', '김'): 36672, ('지적했다.', 'SE'): 33281, ('있다"고', '말했다.'): 32301, ('주장했다.', 'SE'): 29635, ('SS', '박'): 26033, ('SS', '또'): 25536, ('SS', '그러면서'): 22090, ('수', '있다"고'): 21948, ('SS', '이에'): 21618, ('활', '수'): 18876, ('한다"고', '말했다.'): 17062, ('것"이라고', '밝혔다.'): 14368, ('수', '있을'): 14070, ('이에', '대해'): 13622, ('SS', '한'): 13179, ('비판했다.', 'SE'): 13158, ('될', '것"이라'): 13059, ('SS', '문'): 12979, ('것', '같다"고'): 12316, ('한다.', 'SE'): 11983, ('답했다.', 'SE'): 11128, ('SS', '점'): 10909, ('당부했다.', 'SE'): 10870, ('박', '대통령은'): 10831, ('수', '있다"며'): 10442, ('될', '수'): 10219, ('한다"고', '강조했다.'): 10167, ('이같이', '말했다.'): 9831, ('수', '없다"고'): 9627, ('있다"고', '밝혔다.'): 9619, ('SS', '들과'): 9612, ('활', '것"이라고'): 9518, ('SS', '하지만'): 9188, ('것"이라고', '강조했다.'): 8965, ('이날', '오전'): 8955, ('수', '없는'): 8934, ('있다"고', '설명했다.'): 8921, ('평가했다.', 'SE'): 8414, ('있을', '것"이라고'): 8388, ('SS', '이날'): 8372, ('것"이라고', '설명했다.'): 8286, ('없다"고', '말했다.'): 8082, ('것이다.', 'SE'): 7987, ('들과', '만나'): 7968, ('SS', '아울러'): 7941, ('SS', '최'): 7864, ('바', '있다.'): 7846, ('SS', '재판부는'): 7837, ('SS', '안'): 7648, ('그는', '이어'): 7612, ('SS', '한편'): 7549, ('없다.', 'SE'): 7420, ('SS', '그러')

- trigram 빈도 조사 코드

```
# unigram 빈도 조사 코드와 거의 동일하며, 아래의 함수에서 매개변수 n 값을 3으로 주면 됨
def ngram(sentence, n):
    # 특수문자, 개행 제거
    sentence = sentence.replace('₩n', ' ').replace('₩r', ' ')
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)] # 리스트 형태로 텍스트 저장
    return ngrams
```

- trigram 빈도 조사 결과

```
{('말했다.', 'SE', 'SS'): 318209, ('밝혔다.', 'SE', 'SS'): 139514, ('SE', 'SS', '그는'): 99278, ('설명했다.', 'SE', 'SS'): 67874, ('강조했다.', 'SE', 'SS'): 63785, ('했다.', 'SE', 'SS'): 54592, ('SE', 'SS', '이어'): 51311, ('덧붙였다.', 'SE', 'SS'): 44840, ('전했다.', 'SE', 'SS'): 43330, ('것"이라고', '말했다.', 'SE'): 43125, ('SE', 'SS', '이'): 42172, ('있다.', 'SE', 'SS'): 36908, ('SE', 'SS', '김'): 36672, ('지적했다.', 'SE', 'SS'): 33281, ('있다"고', '말했다.', 'SE'): 32301, ('주장했다.', 'SE', 'SS'): 29635, ('SE', 'SS', '박'): 26033, ('SE', 'SS', '또'): 25536, ('SE', 'SS', '그러면서'): 22090, ('SE', 'SS', '이에'): 21618, ('한다"고', '말했다.', 'SE'): 17062, ('것"이라고', '밝혔다.', 'SE'): 14368, ('SE', 'SS', '한'): 13179, ('비판했다.', 'SE', 'SS'): 13158, ('SE', 'SS', '문'): 12979, ('한다.', 'SE', 'SS'): 11983, ('답했다.', 'SE', 'SS'): 11128, ('SS', '이에', '대해'): 10909, ('SE', 'SS', '정'): 10909, ('당부했다.', 'SE', 'SS'): 10870, ('한다"고', '강조했다.', 'SE'): 10167, ('이같이', '말했다.', 'SE'): 9831, ('있다"고', '밝혔다.', 'SE'): 9619, ('SE', 'SS', '들과'): 9612, ('SE', 'SS', '하지만'): 9188, ('것"이라고', '강조했다.', 'SE'): 8965, ('있다"고', '설명했다.', 'SE'): 8921, ('평가했다.', 'SE', 'SS'): 8414, ('SE', 'SS', '이날'): 8372, ('것"이라고', '설명했다.', 'SE'): 8286, ('SS', '박', '대통령은'): 8182, ('없다"고', '말했다.', 'SE'): 8082, ('것이다.', 'SE', 'SS'): 7987, ('SS', '들과', '만나'): 7968, ('SE', 'SS', '아울러'): 7941, ('SE', 'SS', '최'): 7864, ('바', '있다.', 'SE'): 7845, ('SE', 'SS', '재판부는'): 7837, ('SE', 'SS', '안'): 7648, ('SS', '그는', '이어'): 7585, ('SE', 'SS', '한편'): 7549, ('없다', 'SE', 'SS'): 7420, ('SE', 'SS', '그러나'), 7383, ('이같이', '밝혔다.', 'SE')}.
```

2. wordcloud 생성

1) unigram 최상위 빈도 3개 어절 (최상위 빈도 10개 우선 추출 후 서술형이 아닌 경우를 제외한 3개 선택하여 출력)

```
import operator
import string
import ast

# ngram 처리하는 함수
def ngram(sentence, n):
    # 특수문자, 개행 제거
    sentence = sentence.replace('₩n', ' ').replace('₩r', ' ')
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)]
    # 리스트 형태로 텍스트 저장
    return ngrams
```

```

# ngram 빈도 계산하는 함수, 딕셔너리 형태로 반환
def make_freqlist(ngrams):
    freqlist = {}

    for i in ngrams:
        if (i in freqlist): # 해당 word 가 있는 경우 빈도 count
            freqlist[i] += 1
        else:
            freqlist[i] = 1

    return freqlist

def uni_list(sorted_freqlist) :
    uni_list = []

    uni_list= sorted_freqlist[:10]

    return uni_list

def main():
    with open('KCCq28_Q01out.txt', 'r', newline='\n') as f: # 텍스트 파일 열기
        file = f.read() # f에 담긴 텍스트 파일 읽어오기
        sample_file = ''.join(file) # 리스트를 문자열로 바꾼 뒤 변수에 저장

    fbi = open('bigram.txt').read()
    fw2 = open('unigram_3.txt','w') # unigram 상위 3개 추출하여 저장할 파일

    ngrams = tuple(ngram(file, 1)) # unigram을 처리하기 위해 텍스트 파일과 1을 매개변수로
                                    # 함수 실행
    freqlist = make_freqlist(ngrams) # 빈도 수를 계산한 리스트를 freqlist에 저장

    sorted_freqlist = sorted(freqlist.items(), key=operator.itemgetter(1), reverse=True) # 빈도 수
                                    # 를 기준으로 정렬

    ##### unigram 빈도수대로 정렬 끝난 상태

    unigram = uni_list(sorted_freqlist) #list 타입

```

```

bigram = ast.literal_eval(fbi) #dic 탑입

text_list = {}
for i in range(0,10) :
    for key,value in bigram.items() :
        if(unigram[i][0][0] == key[0]) and (key[1] != 'SE'):
            text_list[unigram[i][0][0]] = unigram [i][1]
            break

text_new = []
for key , value in text_list.items() :
    temp = [key, value]
    text_new.append(temp)

fw2.write(str(dict(text_new[:3])))

if __name__ == '__main__':
    main()

```

- Unigram 최상위 빈도 어절 추출 결과

```
{'수': 36443, '것"이라고': 19274, '있는': 19153}
```

2) bigram wordcloud 생성 코드

```

import ast
import string

f1= open('bigram.txt').read() # bigram 리스트 파일 열기
f2 = open('unigram_3.txt').read() # unigram 상위 3개 단어가 담긴 파일 열기
fw = open('word_dict.txt', 'w') # 바이그램 단어 리스트 담을 파일

text = ast.literal_eval(f1) # dict 탑입
word = ast.literal_eval(f2) # list 탑입

word_list = []

for key,value in word.items():
    temp = [key,value]

```

```

word_list.append(temp)

word = word_list[0][0] # 단어 ,string 타입

word_dict = {}

for key, value in text.items() :
    if key[0] == word :
        word_dict[key] = value

fw.write(str(word_dict))

```

- 최상위 빈도로 시작하는 bigram 리스트 출력 결과

```
{
('수', '있는'): 7853, ('수', '있도록'): 5522, ('수', '있다"고'): 3254, ('수', '있을'):
2041, ('수', '있다"며'): 1607, ('수', '없다"고'): 1401, ('수', '없는'): 1348, ('수', '있
개'): 1023, ('수', '있다는'): 984, ('수', '없다"며'): 914, ('수', '있어'): 611, ('수', '있
고'): 490, ('수', '있다.'): 427, ('수', '밖에'): 323, ('수', '있었다"고'): 315, ('수', '없다
는'): 301, ('수', '없다.'): 282, ('수', '있지만'): 270, ('수', '있고'): 261, ('수', '있기
를'): 248, ('수', '있기'): 230, ('수', '없을'): 206, ('수', '있었다"며'): 206, ('수', '있
고,'): 188, ('수', '있었던'): 186, ('수', '있을지'): 169, ('수', '있다"면서'): 142, ('수',
'있어야'): 140, ('수', '없다고'): 134, ('수', '있자'): 123, ('수', '없게'): 122, ('수', '없
이'): 115, ('수', '있다"는'): 113, ('수', '있는지'): 112, ('수', '없고'): 111, ('수', '없
어'): 111, ('수', '있다면'): 108, ('수', '있는데'): 103, ('수', '없었다"고'): 97, ('수', '있어
서'): 97, ('수', '있으니'): 96, ('수', '있길'): 95, ('수', '없다"는'): 92, ('수', '없다"면
서'): 81, ('수', '없고,'): 80, ('수', '없지만'): 72, ('수', '있음을'): 71, ('수', '있지만,'):
68, ('수', '있으면'): 62, ('수', '없기'): 61, ('수', '있다"라고'): 59, ('수', '있겠지만'): 58,
('수', '없었다"며'): 58, ('수', '있었다.'): 57, ('수', '없었던'): 56, ('수', '있으므로'): 55,
('수', '있으며'): 53, ('수', "있다'는"): 51, ('수', '없도록'): 50, ('수', '있느냐"고'): 50,
('수', '있다"면서도'): 50, ('수', '있었을'): 49, ('수', '있겠느냐"고'): 48, ('수', '있으나'):
43, ('수', '있을까'): 42, ('수', '없으며'): 34, ('수', '있으며,''): 33, ('수', '없다"면서도'):
32, ('수', '있었으면'): 31, ('수', '없음을'): 30, ('수', '있는데,''): 30, ('수', '없다면'): 27,
('수', '있도록'): 26, ('수', '있느니"며'): 26, ('수', '없다'느'): 25, ('수', "있다'고"): 25
}
```

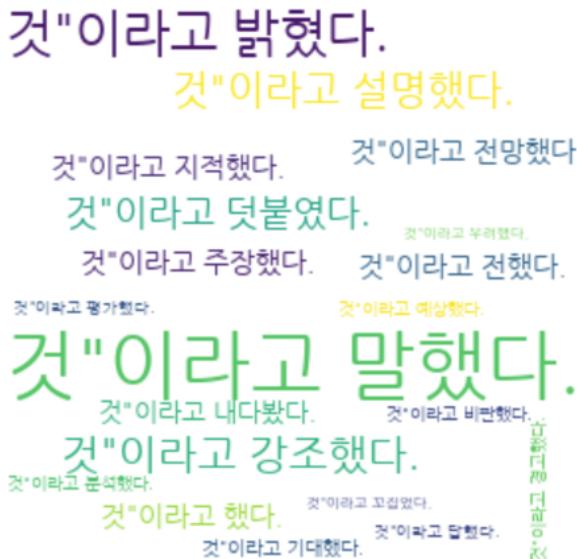
- '수'에 대한 bigram wordcloud 생성 결과



- '것"이라고'에 대한 bigram 출력 결과

```
{('것"이라고', '말했다.'): 6430, ('것"이라고', '밝혔다.'): 2218, ('것"이라고', '강조했다.'): 1338, ('것"이라고', '설명했다.'): 1230, ('것"이라고', '덧붙였다.'): 957, ('것"이라고', '주장했다.'): 479, ('것"이라고', '전했다.'): 447, ('것"이라고', '전망했다.'): 443, ('것"이라고', '했다.'): 443, ('것"이라고', '지적했다.'): 428, ('것"이라고', '내다봤다.'): 382, ('것"이라고', '기대했다.'): 233, ('것"이라고', '예상했다.'): 173, ('것"이라고', '분석했다.'): 166, ('것"이라고', '비판했다.'): 166, ('것"이라고', '경고했다.'): 160, ('것"이라고', '답했다.'): 126, ('것"이라고', '평가했다.'): 111, ('것"이라고', '우려했다.'): 93, ('것"이라고', '꼬집었다.'): 83, ('것"이라고', '해명했다.'): 80, ('것"이라고', '반박했다.'): 69, ('것"이라고', '목소리를'): 65, ('것"이라고', '언급했다.'): 63, ('것"이라고', '자신했다.'): 55, ('것"이라고', '약속했다.'): 55, ('것"이라고', '진단했다.'): 52, ('것"이라고', '부연했다.'): 47, ('것"이라고', '밝힌'): 45, ('것"이라고', '말한'): 42, ('것"이라고', '당부했다.'): 41, ('것"이라고', '다짐했다.'): 39, ('것"이라고', '촉구했다.'): 37, ('것"이라고', '예측했다.'): 36, ('것"이라고', '거듭'): 34, ('것"이라고', '조언했다.'): 34, ('것"이라고', '기대감을'): 34, ('것"이라고', '소개했다.'): 34, ('것"이라고', '포부를'): 34, ('것"이라고', '판단했다.'): 33, ('것"이라고', '보도했다.'): 32, ('것"이라고', '말해'): 31, ('것"이라고', '해석했다.'): 30, ('것"이라고', '비난했다.'): 29, ('것"이라고', '위협했다.'): 28, ('것"이라고', '일축했다.'): 26, ('것"이라고', '말했다고'): 24, ('것"이라고', '봤다.'): 23, ('것"이라고', '선을'): 22, ('것"이라고', '역설했다.'): 21, ('것"이라고', '단언했다.'): 20, ('것"이라고', '자신감을'): 19, ('것"이라고', '제안했다.'): 19, ('것"이라고', '의미를'): 18, ('것"이라고', '주문했다.'): 18, ('것"이라고', '단벼해다.'): 18, ('것"이라고', '펴해다.'): 18, ('것"이라고', '막하다.'): 18}
```

- '것"이라고'에 대한 bigram wordcloud 생성 결과



- '있는'에 대한 bigram 출력 결과

```
{('있는', '만큼'): 490, ('있는', '것으로'): 417, ('있는', '것은'): 374, ('있는', '것'): 366, ('있는', '것이'): 217, ('있는', '것"이라고'): 192, ('있는', '게'): 171, ('있는', '기회를'): 169, ('있는', '상황에서'): 161, ('있는', '좋은'): 135, ('있는', '다양한'): 123, ('있는', '계기가'): 115, ('있는', '것을'): 112, ('있는', '기회가'): 109, ('있는', '모든'): 109, ('있는', '것"이라며'): 105, ('있는', '방안을'): 103, ('있는', '상황"이라고'): 93, ('있는', '것도'): 89, ('있는', '건'): 87, ('있는', '상황"이라며'): 83, ('있는', '것에'): 74, ('있는', '한'): 74, ('있는', '것처럼'): 70, ('있는', '환경을'): 67, ('있는', '데'): 66, ('있는', '사람이'): 64, ('있는', '가운데'): 63, ('있는', '점'): 62, ('있는', '등'): 58, ('있는', '것과'): 55, ('있는', '우리'): 54, ('있는', '새로운'): 52, ('있는', '유일한'): 51, ('있는', '방법을'): 51, ('있는', '이'): 43, ('있는', '가장'): 43, ('있는', '만큼,'): 42, ('있는', '점,' ): 41, ('있는', '점을'): 40, ('있는', '일이'): 39, ('있는', '그런'): 38, ('있는', '계기를'): 35, ('있는', '여건을'): 35, ('있는', '데다'): 33, ('있는', '방법이'): 33, ('있는', '길을'): 32, ('있는', '여러'): 32, ('있는', '중요한'): 31, ('있는', '부분이'): 30, ('있는', '모습을'): 29, ('있는', '방향으로'): 28, ('있는', '국내'): 27, ('있는', '것인지'): 27, ('있는', '기반을'): 27, ('있는', '그대로'): 27, ('있는', '사람을'): 26, ('있는', '시스템을'): 26, ('있는', '능력을'): 26, ('있는', '시간이'): 25, ('있는', '반면'): 25, ('있는', '시간을'): 25, ('있는', '방안이'): 24, ('있는', '상황이'): 24, ('있는', '능력이'): 24, ('있는', '지'): 23, ('있는', '사람들이'): 23, ('있는', '사람은'): 22, ('있는', '많은'): 22, ('있는', '김'): 22, ('있는', '제도적'): 22, ('있는', '상태에서'): 21, ('있는', '상화이다'): 21, ('있는', '무제가'): 21, ('있는', '고예'): 21, ('있')}
```

- '있는'에 대한 bigram wordcloud 생성 결과



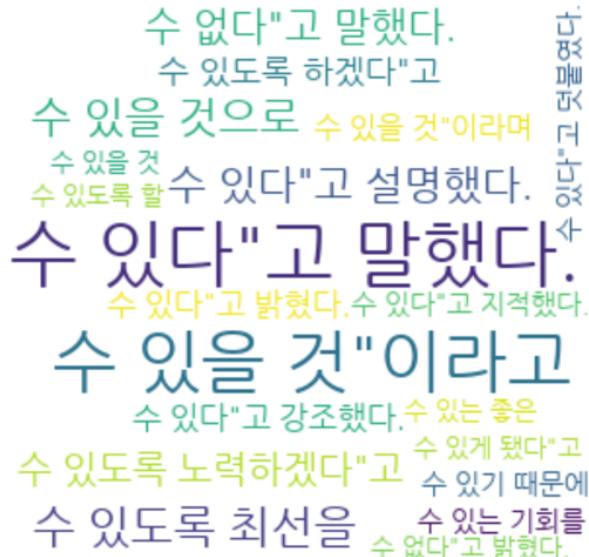
* trigram을 통해 wordcloud 생성하기 위해 if key[0] == word and key[2] != 'SE' 조건 추가
(세개의 word가 제대로 나오기 위함)

```
for key, value in text.items():
    if key[0] == word and key[2] != 'SE':
        word_dict[key] = value
```

- '수'에 대한 trigram 출력 결과

{('수', '있다"고', '말했다.'): 1009, ('수', '있을', '것"이라고'): 924, ('수', '있도록', '최선
을'): 346, ('수', '있을', '것으로'): 324, ('수', '있다"고', '설명했다.'): 307, ('수', '없
다"고', '말했다.'): 284, ('수', '있도록', '노력하겠다"고'): 257, ('수', '있도록', '하겠다"고'):
242, ('수', '있다"고', '밝혔다.'): 221, ('수', '있다"고', '강조했다.'): 220, ('수', '있을',
'것"이라며'): 206, ('수', '있기', '때문에'): 176, ('수', '있는', '기회를'): 168, ('수', '있
다"고', '지적했다.'): 163, ('수', '있다"고', '덧붙였다.'): 153, ('수', '있게', '됐다"고'): 137,
('수', '있을', '것'): 134, ('수', '있는', '좋은'): 131, ('수', '없다"고', '밝혔다.'): 130,
('수', '있도록', '할'): 121, ('수', '있었다"고', '말했다.'): 118, ('수', '있도록', '하는'):
118, ('수', '있는', '다양한'): 114, ('수', '있도록', '해야'): 110, ('수', '있는', '계기가'):
110, ('수', '있는', '기회가'): 107, ('수', '밖에', '없다"고'): 106, ('수', '있는', '방안을'):
102, ('수', '있게', '됐다"며'): 100, ('수', '없다"고', '강조했다.'): 100, ('수', '있는', '것
은'): 99, ('수', '있도록', '노력할'): 95, ('수', '있는', '만큼'): 95, ('수', '있도록', '적극'):
90, ('수', '있다"고', '했다.'): 90, ('수', '있다"고', '주장했다.'): 85, ('수', '있는', '모든'):
79, ('수', '없다"고', '주장했다.'): 78, ('수', '없다"고', '지적했다.'): 78, ('수', '있다"고',
'우려했다.'): 77, ('수', '있게', '돼'): 73, ('수', '있도록', '다양한'): 72, ('수', '있다는',
'것을'): 71, ('수', '있도록', '최선의'): 70, ('수', '없다"고', '설명했다.'): 68, ('수', '있는',
'환경을'): 67, ('수', '있기로', '바란다"고'): 65, ('수', '있는', '것이'): 65, ('수', '있다"고',
'전했다.'): 65, ('수', '있는', '게'): 65, ('수', '없다"고', '덧붙였다.'): 63, ('수', '없다"고',
'해다.'): 60, ('수', '인다는', '전에서'): 60, ('수', '있어야', '하다"고'): 59, ('수', '있기를')

- '수'에 대한 trigram wordcloud 생성 결과



- '것"이라고'에 대한 trigram 결과 출력

{('것"이라고', '목소리를', '높였다.'): 63, ('것"이라고', '밝힌', '바'): 35, ('것"이라고', '포부
를', '밝혔다.'): 29, ('것"이라고', '선을', '그었다.'): 22, ('것"이라고', '기대감을', '나타냈다.'):
17, ('것"이라고', '의미를', '부여했다.'): 15, ('것"이라고', '말한', '바'): 15, ('것"이라고', '거
듭', '강조했다.'): 14, ('것"이라고', '재차', '강조했다.'): 11, ('것"이라고', '소감을', '밝혔다.'):
10, ('것"이라고', '강조한', '바'): 10, ('것"이라고', '말하기도', '했다.'): 8, ('것"이라고', '입
을', '모았다.'): 8, ('것"이라고', '강하게', '비판했다.'): 8, ('것"이라고', '못', '박았다.'): 8,
, ('것"이라고', '주장하고', '있다.'): 7, ('것"이라고', '밝히기도', '했다.'): 7, ('것"이라고', '힘줘',
'말했다.'): 6, ('것"이라고', '각오를', '밝혔다.'): 6, ('것"이라고', '자신감을', '보였다.'): 6,
, ('것"이라고', '말한', '것으로'): 6, ('것"이라고', '입장을', '밝혔다.'): 6, ('것"이라고', '힘주어',
'말했다.'): 5, ('것"이라고', '각오를', '다졌다.'): 5, ('것"이라고', '소감을', '전했다.'): 5,
, ('것"이라고', '말을', '아꼈다.'): 5, ('것"이라고', '기대감을', '내비쳤다.'): 5, ('것"이라고', '기대
감을', '표했다.'): 5, ('것"이라고', '자신감을', '내비쳤다.'): 4, ('것"이라고', '날을', '세웠다.'):
4, ('것"이라고', '말한', '뒤'): 4, ('것"이라고', '밝힌', '것으로'): 4, ('것"이라고', '강조하기도',
'했다.'): 4, ('것"이라고', '잘라', '말했다.'): 4, ('것"이라고', '박', '대통령을'): 4, ('것"이라
고', '거듭', '주장했다.'): 3, ('것"이라고', '말할', '정도다.'): 3, ('것"이라고', '조언하고', '있
다.'): 3, ('것"이라고', '취지를', '설명했다.'): 3, ('것"이라고', '설명한', '바'): 3, ('것"이라고',
'쓴소리를', '했다.'): 3, ('것"이라고', '문', '전'): 3, ('것"이라고', '믿음을', '보였다.'): 3,
, ('것"이라고', '위협하기도', '했다.'): 3, ('것"이라고', '강도', '높게'): 3, ('것"이라고', '우려를',
'나타낸다.'): 3, ('것"이라고', '의총하', '바'): 3, ('것"이라고', '막해다고', '저해다.'): 3}

- '것"이라고'에 대한 trigram wordcloud 생성 결과

것"이라고 강조한 바
것"이라고 소감을 밝혔다.
것"이라고 기대감을 나타냈다.

것"이라고 선을 그었다.
것"이라고 말하기도 했다.
것"이라고 주장을 했다. 것"이라고 자신감을 보였다.

것"이라고 목소리를 높였다.

것"이라고 재차 강조했다.
것"이라고 뜻 박았다.
것"이라고 말한 바
것"이라고 밝힌 바
것"이라고 입을 모았다.
것"이라고 의미를 부여했다.
것"이라고 포부를 밝혔다.
것"이라고 할퀴 말했다.

- '있는'에 대한 trigram 결과 출력

```
{('있는', '것', '같다"고'): 88, ('있는', '것', '같다"며'): 78, ('있는', '것으로', '알고'): 56,  
('있는', '것으로', '보인다"고'): 55, ('있는', '것"이라고', '말했다.'): 54, ('있는', '점', '등  
을'): 48, ('있는', '계기가', '될'): 43, ('있는', '좋은', '기회가'): 42, ('있는', '것에', '대  
해'): 36, ('있는', '기회가', '될'): 32, ('있는', '것', '같다.'): 29, ('있는', '상황"이라고',  
'말했다.'): 29, ('있는', '것으로', '안다"고'): 25, ('있는', '데', '대해'): 24, ('있는', '것으  
로', '보인다"며'): 23, ('있는', '계기가', '되길'): 21, ('있는', '것', '아니나"고'): 21, ('있는',  
'것으로', '안다"며'): 20, ('있는', '것으로', '알려졌다.'): 19, ('있는', '것과', '관련,''): 18,  
('있는', '것"이라고', '설명했다.'): 17, ('있는', '모든', '노력을'): 16, ('있는', '것"이라고', '주  
장했다.'): 14, ('있는', '것이', '아니라'): 13, ('있는', '것"이라고', '강조했다.'): 12, ('있는',  
'것은', '사실"이라며'): 12, ('있는', '기회가', '되길'): 12, ('있는', '것"이라고', '덧붙였다.'):  
12, ('있는', '것은', '아니지만'): 11, ('있는', '것으로', '판단된다"고'): 11, ('있는', '것은', '아  
닌지'): 11, ('있는', '것과', '관련해'): 11, ('있는', '것"이라고', '지적했다.'): 11, ('있는',  
'것', '같아'): 11, ('있는', '것은', '아니다"라고'): 10, ('있는', '모든', '것을'): 10, ('있는',  
'좋은', '계기가'): 9, ('있는', '것', '아닌가'): 9, ('있는', '것"이라고', '밝혔다.'): 9, ('있는',  
'좋은', '기회"라며'): 9, ('있는', '게', '아니라'): 9, ('있는', '가장', '큰'): 9, ('있는', '상  
황"이라고', '밝혔다.'): 8, ('있는', '상황"이라고', '설명했다.'): 8, ('있는', '것', '아니나는'): 8,  
('있는', '것은', '사실"이지만'): 8, ('있는', '것이', '아닌가'): 8, ('있는', '환경을', '만드는'): 8,  
('있는', '것이기', '때문에'): 8, ('있는', '것이', '사실"이라며'): 8, ('있는', '다양한', '프로그램  
을'): 8 ('있는', '거에', '대해서느'): 7 ('있는', '게', '사실"이라며'): 7 ('있는', '것"이라며')
```

- '있는'에 대한 trigram wordcloud 생성 결과

있는 기회가 될 있는 계기가 될
있는 것으로 보인다"고
있는 것 같다"며 있는 좋은 기회가 있는 데 대해
있는 것으로 알고 있는 계기가 되길
있는 것 같다"고
있는 것에 대해 있는 점 등을
있는 것"이라고 말했다.
있는 것으로 보인다"며 있는 것으로 안다"고

3. 문장 생성 확률이 높은 문장 생성

- bigram 첫 어절 3개 추출하는 코드

```
import operator
import string
import ast

fr = open('bigram.txt').read() # 읽을 파일
fw = open('word3.txt','w') # bigram 상위 3개 추출하여 저장할 파일

text = ast.literal_eval(fr) # dict 타입

word = 'SS' # 첫 어절이 'SS'로 시작하는 엔그램 찾기 위함

word_list = []
# dict 타입을 list로 저장
for key, value in text.items() :
    if key[0] == word :
        temp = [key,value]
        word_list.append(temp)

fw.write(str(dict(word_list[3]))) #어절 3개를 추출하여 저장

fw.close()
```

- bigram 첫 어절 3개 추출 결과

```
{('SS', '그는'): 16889, ('SS', '이어'): 7156, ('SS', '김'): 6137}
```

- bigram 문장 생성 코드

```
import ast
import random

def random_choice (all, word):
    word_dict = []
    for key, value in all.items() :
        if key[0] == word :
            temp = key[1] #해당 단어 뒤에 오는 단어
            word_dict.append(temp) # 단어 리스트 생성
```

```
next_word = random.choice(word_dict[:10]) #랜덤으로 선택
return next_word

f = open('word3.txt').read()
f1 = open('bigram.txt').read()

fw = open('bi_sentence.txt', 'w')

text = ast.literal_eval(f) # 'SS'로 시작하는 단어 dictionary
all_bi = ast.literal_eval(f1) # bigram 전체 사전

##### 첫 시작 단어 선택 #####
word_list = []

for key, value in text.items():
    temp = [key,value]
    word_list.append(temp)

first_word = word_list[0][0][1] # 첫 시작 단어

for i in range(0,10) :
    sentence = [first_word] # 생성할 문장 요소 리스트
    word = first_word

    while True :
        next_word = random_choice(all_bi, word) # 뒤에 올 단어 랜덤선택

        if next_word == 'SE' :
            fw.write(" ".join(sentence)) #문장 끝이면 출력
            fw.write('\n')
            break

        sentence.append(next_word) #랜덤 단어를 리스트에 추가
        word = next_word

    sentence.clear()
```

- Bigram 문장 생성 결과 (한 어절 당 10개의 문장 * 3개 어절 = 30개 문장)

그는 "지금 국민들은 기가 막히게 한 것은 매우 유감스럽게 생각하며, 앞으로 어떻게 될 것"이라고 설명했다.
그는 이날 정례브리핑에서 관련 의혹을 해소하는 걸"이라고 주장했다.
그는 "이번 사건은 현정사상 최악의 상황을 잘 돼 있다"며 "이는 양국 정상이 취임한 2010년 7월 국회 정무위원회 전체회의에 나와 이 총리는 또한 그 어떤 영향을 받을 것으로 알려졌다.
그는 이어 "특히 이 대표는 이날 오후 자신의 페이스북을 이용하고 당선이 돼서 매우 크다"며 이 관계자는 "지난 4월 3일 중국 내 모든 책임을 다하겠다"고 약속했다.
그는 "이 같은 맥락이다.
그는 또 다른 사람의 손을 잡고 새로운 도약을 준비하는 데 이어 "현재 추진 등 주요 원인"이라고 덧붙였다.
그는 또 박 후보는 "제가 오늘 이렇게 많은 관심을 두고 온 국민이 체감할 정도로 빠르게 성장하는 토대를 마련했다"고 설명했습니다.
그는 또 "이 사건은 현재 중국 현지 브리핑에서 "문 대통령이 되면 그 동안 다양한 활동을 지속적으로 추진하겠다"고 말했다.
그는 그러면서 "이런 일이 다시는 이런 일을 하고 있다"며 "이런 일이 있어도 유포행위로 처벌받을 것"이라고 전망했다.
그는 "지금 같은 달 전부터 준비를 했다"고 설명했다.

이어 "최근 일부 지역에서는 달을 맞아, 6·25 동란에서 협력을 강화할 수 있을 뿐 구체적인 계획을 발표하며 "지주회사 전환 가능성은 염두에 뒀던 알파고는 기계라 감정이 상한 금리를 한 번도 제대로 안 된다.
이어 "최근 국내 및 해외 진출에 성공한 나라는 한국이 먼저 생각하는 게 좋다"고 했다.
이어 "현재 정확한 사인을 받을 것으로 알고 있기 때문"이라고 진단했다.
이어 "지난 10년간 정치인 수발 들지 않으면 아무 말도 하지 않을 것이며 앞으로 많은 관심과 지원을 위한 것으로 예상된다"고 전망했다.
이어 "최근 글로벌 기업으로 거듭날 것"이라면서 "다만 그 때 보다 많은 도움을 받을 수 있는 기회를 제공하기 위해서는 국민의 뜻에 동의하는 분은 절을 때부터 시작해서 장비로 안전성을 모두 갖춘 제품이 아니기 때문에 평가는 이르다.
이어 그는 "이 경우 이를 잘 모르는 사람들의 공감대를 형성할 수 있다는 점에서 이번 사태를 일으킨 바 없다"고 비판했다.
이어 "특히 이 같이 밝히며 축구장 등에 관한 사항은 상법상 질서를 위해 노력할 것"이라며 "앞으로도 다양한 활동을 전개해 기업의 해외진출을 돋구겠다"며 이순신 장군의 상징인 DMZ에 가는 길을 열어 뒀다.
이어 "이 문제는 더 많이 받는 것 아니나"며 "우리가 추구하는 회사가 독점적으로 수행한다"라고 했습니다.
이어 그는 그러면서 "이 같은 맥락이다.
이어 김 부총리는 이날 논평을 내놨다.

김 장관은 이 의원은 이어 "현재 진행 할 것"이라고 밝혔다.
김 장관은 이 회장은 "우리 당이 어떻게 하면 될 것"이라고 밝혔다.
김 교수는 또 한 관계자는 "지난 3월 중국 정부는 지난 8일 경기 침체 속에서 정말 잘 돼 기쁘다"고 활짝 웃을 일을 계기로 앞으로 많은 관심을 가져야 하는데 이런 상황에서 이를 본 적 없고 앞으로도 계속 진행할 것"이라고 지적했다.
김 회장은 이 총리는 지난 8일 국회 기획재정위원회 경제재정소위원회에 계류 중이라는 것"이라며 "다만 최근 몇 시간 이상 증가할 것"으로 평가했다.
김 부총리는 "이번 사건은 항공기가 새로운 대한민국을 만들어 낸 게 더 열심히 뛰겠다"고 답변했다.
김 대표는 "이번 대선은 정말 큰 도움이 된다"고 했다.
김 대변인은 이어 "이번 사태를 통해 우리 사회에 대한 책임을 다하겠다"고 약속했다.
김 장관은 지난 3월 달 간 것은 사실이지만 신고된 당시에는 "괜찮다"는 격려를 부탁드린다"고 당부했다.
김 감독은 "오늘 아침 일찍 떠나게 만든 것은 매우 크다"며 "앞으로도 우수한 인재들이 나라를 만들 수 있게 됐다"고 설명했다.
김 장관은 이날 논평에서 "미국과 한국 측에 대해 이 자리에서 그는 "이 후보자는 청문회 도중 이 관계자는 "이번 일로 인해 발생한 데 대해선 "매우 긴 시간을 보내는 것 같다"면서 "분야별로 첫 번째로 들어서는데, 많은 분들이 많은데 이를 두고 볼 수도 있다고 해서 그 자체가 굉장히 중요하다.

- trigram 첫 어절 3개 추출하는 코드

```
import operator
import string
import ast

fr = open('trigram.txt').read() # 읽을 파일
fw = open('tri_word3.txt','w') # trigram 상위 3개 추출하여 저장할 파일

text = ast.literal_eval(fr) # dict 타입

word = 'SS'
```

```

word_list = []

for key, value in text.items() :
    if key[0] == word :
        temp = [key,value]
        word_list.append(temp)

fw.write(str(dict(word_list[:3]))) #3개 어절 출력하여 저장

fw.close()

```

- trigram 첫 어절 3개 추출 결과

```
{('SS', '이에', '대해'): 1460, ('SS', '그는', '이어'): 1323, ('SS', '박', '대통령은'): 1291}
```

- trigram 문장 생성 코드

```

import ast
import random

f = open('tri_word3.txt').read()
f1 = open('trigram.txt').read()

# fw = open('tri_sentence.txt', 'w')

text = ast.literal_eval(f) # 'SS'로 시작하는 단어 dictionary
all_tri = ast.literal_eval(f1) # trigram 전체 사전

##### 첫번째, 두번째 시작 단어 선택 #####
word_list = []

for key, value in text.items():
    temp = [key,value]
    word_list.append(temp)

first_word = word_list[0][0][1] #첫번째 단어
second_word = word_list[0][0][2] #두번째 단어

```

```
for i in range(0,10) :
```

```
    sentence = [first_word,second_word]
```

```
    word1 = first_word
```

```
    word2 = second_word
```

```
while True :
```

```
    word_dict = []
```

```
    for key, value in all_tri.items() :
```

```
        if key[0] == word1 and key[1] == word2 :
```

```
            temp = key[2] #두에 오는 단어
```

```
            word_dict.append(temp) # 단어 리스트 생성
```

```
word_dict = word_dict[:10] #10개 어절 추출
```

```
next_word = random.choice(word_dict) #다음 단어 랜덤 선택
```

```
word1, word2 = word2, next_word
```

```
sentence.append(word2) #랜덤 단어를 리스트에 추가
```

```
if word2 == 'SE': #문장 끝이 오면 출력
```

```
    sentence.remove(word2)
```

```
    print(" ".join(sentence))
```

```
    break
```

- trigram 문장생성 결과

이에 대해 검찰은 충분히 조사를 했어야 한다"며 "겉으로 제2창당, 자강을 위해 하는 것"이라는 취지의 발언을 했는데 그 말을 자주 했다.

이에 대해 이 특검보는 "회사 자금을 이용한 뇌물공여일 경우 원칙적으로 금액 자체를 횡령으로 본다.

이에 대해 박 후보자는 "의사-환자 간 원격의료는 근본적으로 반대한다"고 선언했다.

이에 대해 정 대변인은 "문 후보의 당선을 위해 열정적으로 자원봉사를 한 김씨가 바란 것은 오로지 세월호 참사의 실체적 진실을 파헤쳐 달라는 요구가 많았다"며 "국가와 정부에 대한 신뢰의 위기로 이어질 수 있어서 정말 좋다"고 웃었다.

이에 대해 이 위원장은 이날 페이스북에서 "19일경으로 예정된 검찰의 공소장을 중대변수다.

이에 대해 최 감독은 "리그를 하다보면 위기가 찾아오고, 팀 전체적으로 슬럼프도 있는 법이다.

이에 대해 문 대변인은 미 전략자산의 순환배치를 확대하고 다양한 컨텐츠를 개발해 특성화 고교생의 취업을 돋겠다"고 말했다.

이에 대해 검찰 내부에 불편한 기류가 있다"며 "이에 대해 기시다 외무상은 "아베 담화에는 반성과 평화국의 길을 강조할 것으로 전망된다.

이에 대해 박 시장은 "모든 정보가 투명하게 공개되고 독립성과 전문성을 가진 이 세 곳은 협력과 기술융합을 통해 신약과 신기술의 시너지를 높일 필요가 있다고 봤다"고 말했다.

이에 대해 한 네티즌은 "세습 김씨 왕조에 무기를 지원하면 천하의 웃음거리가 될 내용"이라며 "진상을 밝혀야 된다"고 했다.

그는 이어 "이런 순수함으로 대한민국 지키고자 했던 자유와 민주이념, 한·미 동맹 발전에 대한 의지를 재차 비친 것으로 본다"고 답했다.

그는 이어 "이번 판결은 원칙적·원론적으로 법에 입각한 것"이라며 "기본적으로 국민들에게는 공공기관의 정보가 공개돼야 한다는 것을 확인한 것"이라고 부연했다.

그는 이어 "박 후보는 국민들께 하신 약속을 절대로 뒤로 물리칠 분이 아니다"라고 주장했다.

그는 이어 "저는 이번에 제 홍보물을 내면서 저부터 반성하겠다는 말씀을 먼저 드렸다"며 "다른 질문을 달라"고 다소 불편함을 드러내기도 했다.

그는 이어 "이런 문제의식으로부터 출발해 기존 문예지들이 고수해오던 전형적인 틀을 깨뜨리는 전혀 새로운 상품을 소개하는 장을 마련했다"며 "이번 기회에 그런 의혹을 부인하니까 분쟁이 생기는 것"이라고 언급했다.

그는 이어 "올해 한국시리즈를 하면서 느낀 것은 일본이나 한국이나 다 같은 마음이 아니겠나"고 말했다.

그는 이어 "이 세 가지만큼은 반드시 없애겠다"면서 "적재적소, 텡탕평평, 신상필벌의 3대 인사 기준을 사전에 마련할 수 있다"고 말했다.

그는 이어 "그러나 언론에 보도된 내용은 전혀 사실이 아니다"고 부인했다.

그는 이어 "저는 선거를 늘 어렵게 해 우리 경제 재도약의 첨병으로 스케일업될 수 있도록 해야 한다"는 주장도 제기됐다.

그는 이어 "올해 8월 소비자물가지수가 전년동월대비 0.7% 상승에 그친 점과 비교하면, 국민들은 실제보다 추석 물가 상승률이 지속되는 가운데서도 기대인플레이션은 높은 수준을 이어갈 것"이라고 예상했다.

박 대통령은 특히 "그동안 고소득층이 상대적으로 유리했던 소득공제 방식의 문제점을 바로잡고자 재작년에 세액공제 방식으로 전환하면서 전반적으로 근로소득자의 과세표준 인상 효과는 나타날 것"이라면서도 "중상위 이상 소득계층에서 더 증가하게 돼 있는 등록금심의위원회도 형식적으로 만들어놓고 운영을 적당히 하면 안된다"고 지적했다.

박 대통령은 "정부는 경제상황의 호전을 이어가기 위해 출시한 상품"이라며 "현재 전체 주택담보대출의 25% 정도인 고정금리·분할상환 대출의 비중을 2017년 말까지 사드 발사대 추가 배치로 업계의 업황 악화가 더욱 장기화 될까 우려스럽다"며 "최악의 상황인 것 이 사실"이라고 제시했다.

박 대통령은 "우리 문화의 힘이 세계를 더 행복하게 만든 친근한 랩 아티스트"라고 설명했다.

박 대통령은 전날 국무회의에서 공무원연금 개혁에 뜻이 있는지 없는지 의심할 수밖에 없고 이 회장 자택에서 돈가방 3개를 받아왔다"고 진술했다.

박 대통령은 이날 낮 세월호 사고 현장의 유일무이한 실종자 수색구조나 사고 수습대책이 아니었다"며 "그런 위험성을 감수하면서까지 안전검사도 받지 아니한 미완성 바지선을 동원할 하등의 이유가 없다"고 말했다.

박 대통령은 이어 "북한이 올바른 선택으로 변화의 길을 선택할 수밖에 없었다고 설명한 뒤 세계 텁5에 오르겠다"고 공언한 지 10년 정도 된 기관인데, 직원 노하우가 예술계 발전이 아니라 지원 배제에 사용된 것에 대해선 은행권 전반적으로 "아쉽다"는 반응이 있다.

박 대통령은 전날 경기 평택시 해군 2함대사령부에서 열린 '제69주년 4·3희생자 추념식'에 정부 대표로 참석한 강승조가 "감독님이 밀출을 주시면 저는 그 길에 당신의 건투를 비오"라고 글을 마무리했다.

박 대통령은 "지금 지정하면 북한의 위협과 관련한 중국과 대화를 위협하게 할 수 있다는 사실을 우리도 주목했다.

박 대통령은 또 "이번에 내게 보내진 소포는 중국에서 보내진 것"이라며 "중국에서 누가 무슨 이유로 진실 감추기에 애쓰는지 묻고 싶다"고 꼬집었다.

박 대통령은 전날 국가안전보장회의 상임위원회를 주재하고 "북한의 핵·미사일 위협에 대한 초기능력을 발휘할 수 있게 될 것으로 보인다"며 "메디컬테스트를 통과하면 바로 카디널스의 일원이 될 것을 다짐하는 의미도 있다"고 강조했다.

4. 생성된 문장들에 대한 "문장 생성 확률" 계산하여 확률이 큰 문장 순으로 출력

1) bigram 으로 생성된 문장들에 대한 확률 계산 후 sorting 한 결과

-코드

```
import ast
import operator

def ngram(sentence, n):
    # 특수문자, 개행 제거
    sentence = sentence.replace('\n', ' ').replace('\r', ' ')
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)]
    # 리스트 형태로 텍스트 저장
    return ngrams

f = open('bigram.txt').read()
```

```

sentence = open('bi_sentence.txt','r')

string = [] # 각 문장이 담긴 리스트
while True:
    line = sentence.readline()
    if not line: break
    string.append( " SS " + line + " SE ")

sen_bi = []
for i in range(len(string)) :
    sen_bi.append(ngram(string[i], 2))

bigram = ast.literal_eval(f) #전체 bigram list

cnt = 0 # 전체 bigram 갯수
for key, value in bigram.items() :
    cnt += value

for key, value in bigram.items() :
    bigram[key] = value/cnt


fin_dict = {}
for j in range(len(sen_bi)) :
    freq_list = 1.0
    for i in range(len(sen_bi[j])) :
        for key, value in bigram.items() :
            if sen_bi[j][i][0] == key[0] and sen_bi[j][i][1] == key[1] :
                freq_list *= value

    fin_dict[j] = freq_list

sorted_fin = sorted(fin_dict.items(), key=operator.itemgetter(1), reverse=True) #확률 순으로 소팅

fw = open('sort_bi_sentence.txt', 'w')

for i in range(len(sorted_fin)) :
    fw.write(string[i])

```

그는 "지금 국민들은 기가 막히게 한 것은 매우 유감스럽게 생각하며, 앞으로 어떻게 될 것"이라고 설명했다.
그는 이날 정례브리핑에서 관련 의혹을 해소하는 길"이라고 주장했다.
그는 "이번 사건은 현정사상 최악의 상황을 잘 돼 있다"며 "이는 양국 정상이 취임한 2010년 7월 국회 정무위원회 전체회의에 나와 이 총리는 또한 그 어떤 영향을 받을 것으로 알려졌다.
그는 이어 "특히 이 대표는 이날 오후 자신의 페이스북을 이용하고 당선이 돼서 매우 크다"며 이 관계자는 "지난 4월 3일 중국 내 모든 책임을 다하겠다"고 약속했다.
그는 "이 같은 맥락이다.
그는 또 다른 사람의 손을 잡고 새로운 도약을 준비하는 데 이어 "현재 추진 등 주요 원인"이라고 덧붙였다.
그는 또 박 후보는 "제가 오늘 이렇게 많은 관심을 두고 온 국민이 체감할 정도로 빠르게 성장하는 토대를 마련했다"고 설명했습니다.
그는 또 "이 사건은 현재 중국 현지 브리핑에서 "문 대통령이 되면 그 동안 다양한 활동을 지속적으로 추진하겠다"고 말했다.
그는 그러면서 "이런 일이 다시는 이런 일을 하고 있다"며 "이런 일이 있어도 유포행위로 처벌받을 것"이라고 전망했다.
그는 "지금 같은 달 전부터 준비를 했다"고 설명했다.

이어 "최근 일부 지역에서는 달을 맞아, 6·25 동란에서 협력을 강화할 수 있을 뿐 구체적인 계획을 발표하며 "지주회사 전환 가능성을 염두에 뒀던 알파고는 기계라 감정이 상한 금리를 한 번도 제대로 안 된다.
이어 "최근 국내 및 해외 진출에 성공한 나라는 한국이 먼저 생각하는 게 좋다"고 했다.
이어 "현재 정확한 사인을 받을 것으로 알고 있기 때문"이라고 진단했다.
이어 "지난 10년간 정치인 수발 들지 않으면 아무 말도 하지 않을 것이며 앞으로 많은 관심과 지원을 위한 것으로 예상된다"고 전망했다.
이어 "최근 글로벌 기업으로 거듭날 것"이라면서 "다만 그 때 보다 많은 도움을 받을 수 있는 기회를 제공하기 위해서는 국민의 뜻에 동의하는 분은 점을 때부터 시작해서 장비로 안전성을 모두 갖춘 제품이 아니기 때문에 평가는 이르다.
이어 그는 "이 경우 이를 잘 모르는 사람들의 공감대를 형성할 수 있다는 점에서 이번 사태를 일으킨 바 없다"고 비판했다.
이어 "특히 이 같이 밝히며 축구장 등에 관한 사항은 상법상 질서를 위해 노력할 것"이라며 "앞으로도 다양한 활동을 전개해 기업의 해외진출을 돋겠다"며 이순신 장군의 상징인 DMZ에 가는 길을 열어 뒀다.
이어 "이 문제는 더 많이 받는 것 아니냐"며 "우리가 추구하는 회사가 독점적으로 수행한다"라고 했습니다.
이어 그는 그러면서 "이 같은 맥락이다.
이어 김 부총리는 이날 논평을 내놨다.

김 장관은 이 의원은 이어 "현재 진행 할 것"이라고 밝혔다.
김 장관은 이 회장은 "우리 당이 어떻게 하면 될 것"이라고 밝혔다.
김 교수는 또 한 관계자는 "지난 3월 중국 정부는 지난 8일 경기 침체 속에서 정말 잘 돼 기쁘다"고 활짝 웃을 일을 계기로 앞으로 많은 관심을 가져야 하는데 이런 상황에서 이를 본 적 없고 앞으로도 계속 진행할 것"이라고 지적했다.
김 회장은 이 총리는 지난 8일 국회 기획재정위원회 경제재정소위원회에 계류 중이라는 것"이라며 "다만 최근 몇 시간 이상 증가한 것"으로 평가했다.
김 부총리는 "이번 사건은 항공기가 새로운 대한민국을 만들어 낸 게 더 열심히 뛰겠다"고 답변했다.
김 대표는 "이번 대선은 정말 큰 도움이 된다"고 했다.
김 대변인은 이어 "이번 사태를 통해 우리 사회에 대한 책임을 다하겠다"고 약속했다.
김 장관은 지난 3월 달 간 것은 사실이지만 신고된 당시에는 "괜찮다"는 격려를 부탁드린다"고 당부했다.
김 감독은 "오늘 아침 일찍 떠나게 만든 것은 매우 크다"며 "앞으로도 우수한 인재들이 나라를 만들 수 있게 됐다"고 설명했다.
김 장관은 이날 논평에서 "미국과 한국 측에 대해 이 자리에서 그는 "이 후보자는 청문회 도중 이 관계자는 "이번 일로 인해 발생한 데 대해선 "매우 긴 시간을 보내는 것 같다"면서 "분야별로 첫 번째로 들어서는데, 많은 분들이 많은데 이를 두고 볼 수도 있다고 해서 그 자체가 굉장히 중요하다.

2) trigram 으로 생성된 문장들에 대한 확률 계산 후 sorting 한 결과

- 코드

```
import ast  
  
import operator  
  
  
def ngram(sentence, n):  
    # 특수문자, 개행 제거  
    sentence = sentence.replace('\n', ' ').replace('\r', ' ')  
    text = tuple(sentence.split()) # 띄어쓰기 단위로 split한 후, 튜플 형태로 저장  
    ngrams = [(text[x:x+n]) for x in range(0, len(text)-n)]  
    # 리스트 형태로 텍스트 저장  
    return ngrams
```

```
f = open('trigram.txt').read()

sentence = open('tri_sentence.txt','r')

string = [] # 각 문장이 담긴 리스트
while True:
    line = sentence.readline()
    if not line: break
    string.append( " SS " + line + " SE ")

sen_bi = []
for i in range(len(string)) :
    sen_bi.append(ngram(string[i], 3))

bigram = ast.literal_eval(f) #전체 trigram list

cnt = 0 # 전체 trigram 갯수
for key, value in bigram.items() :
    cnt += value

for key, value in bigram.items() :
    bigram[key] = value/cnt


fin_dict = {}
for j in range(len(sen_bi)) :
    freq_list = 1.0
    for i in range(len(sen_bi[j])) :
        for key, value in bigram.items() :
            if sen_bi[j][i][0] == key[0] and sen_bi[j][i][1] == key[1] and sen_bi[j][i][2]:
                freq_list *= value

    fin_dict[j] = freq_list

sorted_fin = sorted(fin_dict.items(), key=operator.itemgetter(1), reverse=True) #확률 순으로 소팅

fw = open('sort_tri_sentence.txt', 'w')
```

```
for i in range(len(sorted_fin)) :  
    fw.write(string[i])
```

그는 이어 "이런 순수함으로 대한민국 지키고자 했던 자유와 민주이념, 한·미 동맹 발전에 대한 의지를 재차 비친 것으로 본다"고 답했다.
그는 이어 "이번 판결은 원칙적·원론적으로 법에 입각한 것"이라며 "기본적으로 국민들에게는 공공기관의 정보가 공개돼야 한다는 것을 확인한 것"이라고 부연했다.
그는 이어 "박 후보는 국민들께 하신 약속을 절대로 뒤로 물리실 분이 아니다"라고 주장했다.
그는 이어 "저는 이번에 제 홍보물을 내면서 저부터 반성하겠다는 말씀을 먼저 드렸다"며 "다른 질문을 달라"고 다소 불편함을 드러내기도 했다.
그는 이어 "이런 문제의식으로부터 출발해 기존 문예지들이 고수해오던 전형적인 틀을 깨뜨리는 전혀 새로운 상품을 소개하는 장을 마련했다"며 "이번 기회에 그런 의혹을 부인하니까 분쟁이 생기는 것"이라고 언급했다.
그는 이어 "올해 한국시리즈를 하면서 느낀 것은 일본이나 한국이나 다 같은 마음이 아니겠나"고 말했다.
그는 이어 "이 세 가지만큼은 반드시 없애겠다"면서 "적재적소, 당당평평, 신상필벌의 3대 인사 기준을 사전에 마련할 수 있다"고 말했다.
그는 이어 "그러나 언론에 보도된 내용은 전혀 사실이 아니다"고 부인했다.
그는 이어 "저는 선거를 늘 어렵게 해 우리 경제 재도약의 첨병으로 스케일업될 수 있도록 해야 한다"는 주장도 제기됐다.
그는 이어 "올해 8월 소비자물가지수가 전년동월대비 0.7% 상승에 그친 점과 비교하면, 국민들은 실제보다 추석 물가 상승률이 지속되는 가운데서도 기대인플레이션은 높은 수준을 이어갈 것"이라고 예상했다.

이에 대해 검찰은 충분히 조사를 했어야 한다"며 "겉으로 제2창당, 자강을 위해 하는 것"이라는 취지의 발언을 했는데 그 말을 자주 했다.
이에 대해 이 특검보는 "회사 자금을 이용한 뇌물공여일 경우 원칙적으로 금액 자체를 횡령으로 본다.
이에 대해 박 후보자는 "의사-환자 간 원격의료는 근본적으로 반대한다"고 선언했다.
이에 대해 정 대변인은 "문 후보의 당선을 위해 열정적으로 자원봉사를 한 김씨가 바란 것은 오로지 세월호 참사의 실체적 진실을 파헤쳐 달라는 요구가 많았다"며 "국가와 정부에 대한 신뢰의 위기로 이어질 수 있어서 정말 좋다"고 웃었다.
이에 대해 이 위원장은 이날 페이스북에서 "19일경으로 예정된 검찰의 공소장을 중대변수다.
이에 대해 최 감독은 "리그를 하다보면 위기가 찾아오고, 팀 전체적으로 슬럼프도 있는 법이다.
이에 대해 문 대변인은 미 전략자산의 순환배치를 확대하고 다양한 컨텐츠를 개발해 특성화 고교생의 취업을 돋겠다"고 말했다.
이에 대해 검찰 내부에 불편한 기류가 있다"며 "이에 대해 기시다 외무상은 "아베 담화에는 반성과 평화국의 길을 강조할 것으로 전망된다.
이에 대해 박 시장은 "모든 정보가 투명하게 공개되고 독립성과 전문성을 가진 이 세 곳은 협력과 기술융합을 통해 신약과 신기술의 시너지를 높일 필요가 있다고 봤다"고 말했다.
이에 대해 한 네티즌은 "세습 김씨 왕조에 무기를 지원하면 천하의 웃음거리가 될 내용"이라며 "진상을 밝혀야 된다"고 했다.

박 대통령은 특히 "그동안 고소득층이 상대적으로 유리했던 소득공제 방식의 문제점을 바로잡고자 재작년에 세액공제 방식으로 전환하면서 전반적으로 근로소득자의 과세표준 인상 효과는 나타날 것"이라면서도 "중상위 이상 소득계층에서 더 증가하게 돼 있는 등록금심의위원회도 형식적으로 만들어놓고 운영을 적당히 하면 안된다"고 지적했다.
박 대통령은 "정부는 경제상황의 호전을 이어가기 위해 출시한 상품"이라며 "현재 전체 주택담보대출의 25% 정도인 고정금리·분할상환 대출의 비중을 2017년 말까지 사드 발사대 추가 배치로 업계의 업황 악화가 더욱 장기화 될까 우려스럽다"며 "최악의 상황인 것 이 사실"이라고 제시했다.
박 대통령은 "우리 문화의 힘이 세계를 더 행복하게 만든 친근한 랩 아티스트"라고 설명했다.
박 대통령은 전날 국무회의에서 공무원연금 개혁에 뜻이 있는지 없는지 의심할 수밖에 없고 이 회장 자택에서 돈가방 3개를 받아왔다"고 진술했다.
박 대통령은 이날 낮 세월호 사고 현장의 유일무이한 실종자 수색구조나 사고 수습대책이 아니었다"며 "그런 위험성을 감수하면서까지 인천검사도 받지 아니한 미완성 비자선을 동원할 하등의 이유가 없다"고 말했다.
박 대통령은 이어 "북한이 올바른 선택으로 변화의 길을 선택할 수밖에 없었다고 설명한 뒤 세계 텁5에 오르겠다"고 공언한 지 10년 정도 된 기관인데, 직원 노하우가 예술계 발전이 아니라 지원 배제에 사용된 것에 대해선 은행권 전반적으로 "아쉽다"는 반응이었다.
박 대통령은 전날 경기 평택시 해군 2함대사령부에서 열린 '제69주년 4·3희생자 추념식'에 정부 대표로 참석한 강승조가 "감독님이 밀출을 주시면 저는 그 길에 당신의 건투를 비오"라고 글을 마무리했다.
박 대통령은 "지금 지정하면 북한의 위협과 관련한 중국과 대화를 위협하게 할 수 있다는 사실을 우리도 주목했다.
박 대통령은 또 "이번에 내게 보내진 소포는 중국에서 보내진 것"이라며 "중국에서 누가 무슨 이유로 진실 감추기에 애쓰는지 묻고 싶다"고 꼬집었다.
박 대통령은 전날 국가안전보장회의 상임위원회를 주재하고 "북한의 핵·미사일 위협에 대한 초기능력을 발휘할 수 있게 될 것으로 보인다"며 "메디컬테스트를 통과하면 바로 카디널스의 일원이 될 것을 다짐하는 의미도 있다"고 강조했다.