

Copyright (c) 2019 [윤기태]

<https://github.com/yoontk200/python-data-analysis>
(<https://github.com/yoontk200/python-data-analysis>).

MIT License (<https://github.com/yoontk200/python-data-analysis/blob/master/LICENSE.txt>).

(가제) 파이썬 데이터 분석

2.2) 트위터 API로 연관 키워드 분석하기

바로가기

- [<Step1. API 호출> : 트위터 API로 데이터 가져오기](#)
 - [API 데이터로 데이터 프레임 생성하기]
 - [<Step2. 추출> : 키워드 추출](#)
 - [텍스트 데이터 전처리]
 - [nltk, konlpy를 이용한 키워드 추출]
 - [<Step3. 분석> : 연관 분석을 이용한 키워드 분석](#)
 - [연관 키워드 추출하기]
 - [단어 빈도 추출하기]
 - [<Step4. 시각화> : 연관 키워드 네트워크 시각화](#)
 - [연관 키워드 네트워크 시각화]
-

In [1]:

```
# -*- coding: utf-8 -*-
```

```
# 주피터 노트북을 실행한 브라우저에서 바로 그림을 볼 수 있게끔 만드는 것
```

```
#브라우저 내부(inline)에 바로 그려지도록 해주는 코드
```

```
%matplotlib inline
```

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

<Step1. API 호출> : 트위터 API로 데이터 가져오기

[API 데이터로 데이터 프레임 생성하기]

- API 사용법 참고 : [https://github.com/yoont200/python-data-analysis/blob/master/chapter_text/\(%EC%B0%B8%EA%B3%A0\)%20using-twitter-api.ipynb](https://github.com/yoont200/python-data-analysis/blob/master/chapter_text/(%EC%B0%B8%EA%B3%A0)%20using-twitter-api.ipynb) ([https://github.com/yoont200/python-data-analysis/blob/master/chapter_text/\(%EC%B0%B8%EA%B3%A0\)%20using-twitter-api.ipynb](https://github.com/yoont200/python-data-analysis/blob/master/chapter_text/(%EC%B0%B8%EA%B3%A0)%20using-twitter-api.ipynb))
- 아래 코드 실행을 위해, anaconda prompt 혹은 Terminal에서 아래와 같은 패키지를 설치해 줍니다.
 - (env_name) pip install tweepy
- 혹은 아래의 코드로 라이브러리를 설치합니다.
- [2022.02.04] tweepy 4 버전 설치시 api.search에서 에러 발생 => tweepy 3 버전으로 설치
- pip install tweepy==3.10.0

In [2]:

```
#!/pip install tweepy
# tweepy 3 버전으로 설치
#!/pip install tweepy==3.10.0
```

Requirement already satisfied: tweepy==3.10.0 in c:\users\yj\anac
nda3\lib\site-packages (3.10.0)

Requirement already satisfied: requests[socks]>=2.11.1 in c:\users
\yj\anaconda3\lib\site-packages (from tweepy==3.10.0) (2.27.1)

Requirement already satisfied: six>=1.10.0 in c:\users\yj\anaconda3
\lib\site-packages (from tweepy==3.10.0) (1.16.0)

Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users
\yj\anaconda3\lib\site-packages (from tweepy==3.10.0) (1.3.1)

Requirement already satisfied: oauthlib>=3.0.0 in c:\users\yj\anaco
nda3\lib\site-packages (from requests-oauthlib>=0.7.0->tweepy==
3.10.0) (3.2.0)

Requirement already satisfied: charset-normalizer~=2.0.0 in c:\user
s\yj\anaconda3\lib\site-packages (from requests[socks]>=2.11.1->t
weepy==3.10.0) (2.0.4)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\yj\ana
conda3\lib\site-packages (from requests[socks]>=2.11.1->tweepy==
3.10.0) (2021.10.8)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\yj\ana
conda3\lib\site-packages (from requests[socks]>=2.11.1->tweepy
==3.10.0) (1.26.7)

Requirement already satisfied: idna<4,>=2.5 in c:\users\yj\anacond
a3\lib\site-packages (from requests[socks]>=2.11.1->tweepy==3.10.
0) (3.2)

Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in c:\users\yj
\anaconda3\lib\site-packages (from requests[socks]>=2.11.1->twee
py==3.10.0) (1.7.1)

트위터 API 가져오기

- 트위터의 키워드 크롤링 기능을 사용하기 위해 트위터 앱에서 발급받은 KEY와 TOKEN 정보를 입력
- 총 4가지 정보를 입력하여 코드 실행

- tweepy의 OAuthHandler() 클래스가 자동으로 개인정보 인증 완료

In [3]:

```
import tweepy

# 발급 완료된 키 입력
CONSUMER_KEY = "1guUu16Re83rAexS7VhsCAQv0"
CONSUMER_SECRET = "DMlEHpONvAo5ZPtDt1wZPFrHplrQIYijOGgD2suu2p9HW0zQ"
ACCESS_TOKEN_KEY = "169786640-wXppNaveQqbXzOkf6ls3Nt4QQPs04JQUZkoypljv"
ACCESS_TOKEN_SECRET = "9ubAWb5RKllyREa3jc6NM6JUnLp8RILbieFrmEMRDONt0"

# 개인정보 인증을 요청하는 Handler
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)

# 인증 요청을 수행
auth.set_access_token(ACCESS_TOKEN_KEY, ACCESS_TOKEN_SECRET)

# twitter API를 사용하기 위한 준비
api = tweepy.API(auth)
```

In [4]:

```
# tweepy 버전 출력
tweepy.__version__
```

Out[4]:

'3.10.0'

'손흥민' 키워드 검색

In [5]:

```
# Twitter Developer Platform의 Data dictionary 참조
# https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model

# twitter API를 사용하여 '손흥민' 이 포함된 트윗들을 크롤링한 뒤, 'user_mentions'와 'hashtags'와 'keywords'를 추출
keyword = "손흥민"

# 특정 키워드 크롤링은 search() 함수 수행
tweets = api.search(keyword)

# entities() 함수로 특정 속성의 메타데이터에 접근하여 정보 추출
for tweet in tweets:
    print(tweet.text)
    print(tweet.entities['user_mentions'])
    print(tweet.entities['hashtags'])
    print(tweet.created_at) # 트윗이 생성된 UTC 시간, 예) "2022-02-04 01:47:09"
```

2022-02-04 07:22:27

RT @Son7_Fcthai: Good morning 🌞 Friday

Sonny Photo Action Tumi 🔥🔥🔥

Edit: sonheungmin fcthai

© @tumitravel

Youtube :https://t.co/5jcAUw...

```
[{'screen_name': 'Son7_Fcthai', 'name': 'SonHeungMinFCThai', 'id': 1419538478606340097, 'id_str': '1419538478606340097', 'indices': [3, 15]}, {'screen_name': 'tumitravel', 'name': 'tumitravel', 'id': 24757830, 'id_str': '24757830', 'indices': [99, 110]}]
```

[]

2022-02-04 07:22:19

여자신음

판매

섹스

In [6]:

tweets

Out[6]:

```
[Status(_api=<tweepy.api.API object at 0x000002013E04A640>, _json={'created_at': 'Fri Feb 04 07:25:14 +0000 2022', 'id': 1489500277069971461, 'id_str': '1489500277069971461', 'text': '손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서... 그러니까 느낌적으로 말하면, 손흥민은 쉽지 않은 경기에서는 답을 내는 것 같은데 어려운 경기에서는 답 내는 모습을 본 적이 있나? 없나? 모르겠드아...'}, 'truncated': False, 'entities': {'hashtags': [], 'symbols': [], 'user_mentions': [], 'urls': []}, 'metadata': {'result_type': 'recent', 'iso_language_code': 'ko'}, 'source': '<a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>', 'in_reply_to_status_id': 1489499231673270273, 'in_reply_to_status_id_str': '1489499231673270273', 'in_reply_to_user_id': 76188562, 'in_reply_to_user_id_str': '76188562', 'in_reply_to_screen_name': 'izalcorace', 'user': {'id': 76188562, 'id_str': '76188562', 'name': '보리빵', 'screen_name': 'izalcorace', 'location': '대한민국 서울', 'description': '이제껏의 스연 친구! 보리! 스연!'}])]
```

데이터 프레임 형태로 수집

In [7]:

```
# 크롤링된 데이터를 저장할 데이터 프레임 생성
columns = ['created', 'tweet_text']
df = pd.DataFrame(columns=columns)

# 크롤링을 수행할 갯수를 입력
max_tweets = 1000

# Cursor 객체를 사용하여 크롤링 수행
# Cursor()로 keyword의 값에 해당하는 정보를 items()으로 지정한 갯수만큼 크롤링
searched_tweets = [status for status in tweepy.Cursor(api.search, q=keyword).items(1000)]

# '손흥민'이 포함된 1000개의 트윗들에서, 'text', 'created_at' 정보를 데이터 프레임으로 저장
for tweet in searched_tweets:
    tweet_json = tweet._json
    tweet_text = tweet_json['text']
    created = tweet_json['created_at']
    row = [created, tweet_text]
    series = pd.Series(row, index=df.columns) # created, tweet_text로 시리즈를 생성
    df = df.append(series, ignore_index=True) # 생성된 시리즈를 데이터프레임에 추가
```

In [8]:

```
# 데이터 프레임 상위 5개 출력
df.head()
```

Out[8]:

	created	tweet_text
0	Fri Feb 04 07:25:14 +0000 2022	손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서... 그러니 까 느낌...
1	Fri Feb 04 07:23:21 +0000 2022	RT @Son7_Fcthai: 🇹🇪สเปอर्स ได้ขึ้นอันดับหนึ่งใน...
2	Fri Feb 04 07:22:27 +0000 2022	RT @Son7_Fcthai: Sonny Photo Action Tumi 🍌🍌🍌/2...
3	Fri Feb 04 07:22:19 +0000 2022	RT @Son7_Fcthai: Good morning 🌞 Friday \n\nSo...
4	Fri Feb 04 07:14:16 +0000 2022	여자신을 \n판매 \n섹스 \n자위가슴 \n자위영삼 \n출장만남 \n보지 \n크 리스...

In [9]:

```
# 데이터 프레임을 csv 로 저장  
# df.to_csv("tweet_temp.csv", index=False)
```

<Step2. 추출> : 키워드 추출

[텍스트 데이터 전처리]

- 한글 문자열을 기준으로 키워드 추출 수행
- tweet_text 데이터에서 한글 문자열만을 추출하여 ko_text 생성

In [10]:

```
#df = pd.read_csv("tweet_temp.csv")  
#df.head()
```

In [11]:

```
# 정규표현 라이브러리  
import re  
  
# 텍스트 정제 함수 : 한글 및 띄어쓰기 이외의 문자는 전부 제거  
def text_cleaning(text):  
    hangul = re.compile('[^ㄱ-ㅣ가-힣]+') # 한글의 정규표현식  
    result = hangul.sub("", text)  
    return result
```


In [12]:

```
# 텍스트 정제 함수text_cleaning()를 'tweet_text' 피처에 이를 적용, 한글과 띄어쓰기만 추출
df['ko_text'] = df['tweet_text'].apply(lambda x: text_cleaning(x))
df.head()
```

Out[12]:

	created	tweet_text	ko_text
0	Fri Feb 04 07:25:14 +0000 2022	손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서... 그러니까 느 낌...	손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서 그러니까 느낌적으로...
1	Fri Feb 04 07:23:21 +0000 2022	RT @Son7_Fcthai: 🇹🇭สเปอर्स ได้ขึ้น อันดับหนึ่งใน...	
2	Fri Feb 04 07:22:27 +0000 2022	RT @Son7_Fcthai: Sonny Photo Action Tumi 🍌🍌🍌/2...	
3	Fri Feb 04 07:22:19 +0000 2022	RT @Son7_Fcthai: Good morning ☀️ Friday \n\nSo...	
4	Fri Feb 04 07:14:16 +0000 2022	여자신음 \n판매 \n섹스 \n자위가슴 \n 자위영삼 \n출장만남 \n보지 \n크리 스...	여자신음 판매 섹스 자위가슴 자위영삼 출장만 남 보지 크리스마스조건만남출장만남출장삼 손...

[konlpy를 이용한 키워드 추출]

- 명사만 추출한 후 불용어 및 한글자 키워드 제거

In [13]:

```
from konlpy.tag import Okt
from collections import Counter

# 불용어 사전의 텍스트 문서 읽어 stopwords 리스트 생성
# 한국어 약식 불용어사전 예시 파일, 출처 - (https://www.ranks.nl/stopwords/korean)
korean_stopwords_path = "../data/korean_stopwords.txt"
with open(korean_stopwords_path, encoding='utf8') as f:
    stopwords = f.readlines()
stopwords = [x.strip() for x in stopwords] # strip()으로 공백 및 줄바꿈 제거

# 품사 중 명사만 추출하여 한글자 키워드 및 불용어 제거
# Okt(Open Korean Text)는 트위터에서 만든 오픈소스 한국어 처리기인 twitter-korean-text-processor

def get_nouns(x):
    # 명사만 추출
    nouns_tagger = Okt() # Okt()클래스 선언
    nouns = nouns_tagger.nouns(x)

    # 한글자 키워드 제거
    nouns = [noun for noun in nouns if len(noun) > 1]

    # 불용어를 제거
    nouns = [noun for noun in nouns if noun not in stopwords]

    return nouns
```

In [14]:

```
stopwords
```

Out[14]:

```
['아',  
'휴',  
'아이구',  
'아이쿠',  
'아이고',  
'어',  
'나',  
'우리',  
'저희',  
'따라',  
'의해',  
'을',  
'를',  
'에',  
'시'
```

In [15]:

```
# 'ko_text' 피처에 get_nouns() 함수를 적용하여 불용어 및 한글자를 제거한 명사 추출하여 'nouns'
df['nouns'] = df['ko_text'].apply(lambda x: get_nouns(x))
print(df.shape) # 행과 열의 갯수를 튜플로 반환
df.head()
```

(1000, 4)

Out[15]:

	created	tweet_text	ko_text	nouns
0	Fri Feb 04 07:25:14 +0000 2022	손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서... 그러니까 느낌...	손흥민이 떠오르며 기대할 수 밖 에 없긴 한데 아쉬움도 좀 있어서 그러니까 느낌적으로...	[손흥민, 아쉬움, 느낌, 손 흥민, 경기, 경기, 모습]
1	Fri Feb 04 07:23:21 +0000 2022	RT @Son7_Fcthai: 🇷🇺 เปอร์ส ได้ขึ้นอันดับหนึ่งใน...		[]
2	Fri Feb 04 07:22:27 +0000 2022	RT @Son7_Fcthai: Sonny Photo Action Tumi 🍑🍑 🍑/2...		[]
3	Fri Feb 04 07:22:19 +0000 2022	RT @Son7_Fcthai: Good morning 🌞 Friday \n\nSo...		[]
4	Fri Feb 04 07:14:16 +0000 2022	여자신음 \n판매 \n섹스 \n자 위가슴 \n자위영삼 \n출장만 남 \n보지 \n크리스...	여자신음 판매 섹스 자위가슴 자 위영삼 출장만남 보지 크리스마스 조건만남출장만남출장삼손...	[여자, 신음, 판매, 섹스, 자위, 가슴, 자위, 출장, 만 남, 보지, 크리스마스...

<Step3. 분석> : 연관 분석을 이용한 키워드 분석

[연관 키워드 추출하기]

[연관 규칙]

- <https://ratsgo.github.io/machine%20learning/2017/04/08/apriori/>
(<https://ratsgo.github.io/machine%20learning/2017/04/08/apriori/>)
- 연관규칙 (아프리오리, Apriori 알고리즘)
<https://blog.naver.com/zzz90zzz/221807210555>
(<https://blog.naver.com/zzz90zzz/221807210555>)
- 연관규칙은 비지도학습으로 대규모 거래 데이터로 부터 함께 구매될 규칙을 도출하여 고객이 특정 상품 구매시 이와 연관성이 높은 상품을 추천하는 것
- Apriori 알고리즘 : 간단한 성능 측정치를 이용해 거대한 DB에서 데이터간의 연관성을 찾는 알고리즘

[연관규칙에서 사용하는 3가지 통계척도]

- 1. 지지도 (support) : 특정 아이템이 데이터에서 발생하는 빈도
- 2. 신뢰도 (confidence) : 두 아이템의 연관규칙이 유용한 규칙일 가능성의 척도
- 3. 향상도 (lift) : 두 아이템의 연관 규칙이 우연인지 아닌지를 나타내는 척도
- 아래 코드 실행을 위해, anaconda prompt 혹은 Terminal에서 아래와 같은 패키지들을 설치
 - (env_name) pip install apriori apyori
- 혹은 아래의 코드로 라이브러리를 설치

In [16]:

```
#!pip install apriori apyori
```

연관 분석 연습

In [17]:

연관 분석 연습 1

from apyori **import** apriori

장바구니 형태의 데이터(트랜잭션 데이터)를 생성

```
transactions = [
    ['손흥민', '시소코'],
    ['손흥민', '케인'],
    ['손흥민', '케인', '포체티노']
]
```

연관 분석을 수행

results = list(apriori(transactions))

for result **in** results: **print**(result)

```
RelationRecord(items=frozenset({'손흥민'}), support=1.0, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])
```

```
RelationRecord(items=frozenset({'시소코'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'시소코'}), confidence=0.3333333333333333, lift=1.0)])
```

```
RelationRecord(items=frozenset({'케인'}), support=0.6666666666666666, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'케인'}), confidence=0.6666666666666666, lift=1.0)])
```

```
RelationRecord(items=frozenset({'포체티노'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'포체티노'}), confidence=0.3333333333333333, lift=1.0)])
```

```
RelationRecord(items=frozenset({'시소코', '손흥민'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'시소코', '손흥민'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'손흥민'}), items_add=frozenset({'시소코'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'시소코'}), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])
```

```
RelationRecord(items=frozenset({'손흥민', '케인'}), support=0.666666
```

```

6666666666, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'손흥민', '케인'}), confidence=0.6666666666666666, lift=1.0), OrderedStatistic(items_base=frozenset({'손흥민'}), items_add=frozenset({'케인'}), confidence=0.6666666666666666, lift=1.0), OrderedStatistic(items_base=frozenset({'케인'}), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])
RelationRecord(items=frozenset({'포체티노', '손흥민'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'포체티노', '손흥민'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'손흥민'}), items_add=frozenset({'포체티노'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'포체티노'}), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])
RelationRecord(items=frozenset({'포체티노', '케인'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'포체티노', '케인'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'케인'}), items_add=frozenset({'포체티노'}), confidence=0.5, lift=1.5), OrderedStatistic(items_base=frozenset({'포체티노'}), items_add=frozenset({'케인'}), confidence=1.0, lift=1.5)])
RelationRecord(items=frozenset({'포체티노', '손흥민', '케인'}), support=0.3333333333333333, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'포체티노', '손흥민', '케인'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'손흥민'}), items_add=frozenset({'포체티노', '케인'}), confidence=0.3333333333333333, lift=1.0), OrderedStatistic(items_base=frozenset({'케인'}), items_add=frozenset({'포체티노', '손흥민'}), confidence=0.5, lift=1.5), OrderedStatistic(items_base=frozenset({'포체티노'}), items_add=frozenset({'손흥민', '케인'}), confidence=1.0, lift=1.5), OrderedStatistic(items_base=frozenset({'손흥민', '케인'}), items_add=frozenset({'포체티노'}), confidence=0.5, lift=1.5), OrderedStatistic(items_base=frozenset({'포체티노', '손흥민'}), items_add=frozenset({'케인'}), confidence=1.0, lift=1.5), OrderedStatistic(items_base=frozenset({'포체티노', '케인'}), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])

```

In [18]:

연관 분석 연습 2

지지도 0.5, 신뢰도 0.6, 향상도 1.0 이상이면서 (손흥민, 케인) 처럼 규칙의 크기가 2 이하인 규칙

```
list(apriori(transactions,
             min_support=0.5,
             min_confidence=0.6,
             min_lift=1.0,
             max_length=2))
```

Out[18]:

```
[RelationRecord(items=frozenset({'손흥민'}), support=1.0, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)]),
 RelationRecord(items=frozenset({'케인'}), support=0.6666666666666666, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'케인'}), confidence=0.6666666666666666, lift=1.0)]),
 RelationRecord(items=frozenset({'손흥민', '케인'}), support=0.6666666666666666, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'손흥민', '케인'}), confidence=0.6666666666666666, lift=1.0), OrderedStatistic(items_base=frozenset({'손흥민'}), items_add=frozenset({'케인'}), confidence=0.6666666666666666, lift=1.0), OrderedStatistic(items_base=frozenset({'케인'}), items_add=frozenset({'손흥민'}), confidence=1.0, lift=1.0)])]
```

트위터 연관 키워드 분석

In [19]:

트랜잭션 데이터를 추출

transactions = df['nouns'].tolist()

#transactions = [transaction for transaction in transactions if transaction] # 공백 문자

print(transactions)

'여행', '기록', '리야', '옹이', '조건', '만남'], ['어제', '개미', '대부분', '커뮤',
 '게시판', '초토화', '부들', '거리', '윤석열', '자살골', '이쯤', '손흥민', '푸스
 카스상'], ['여자', '신음', '판매', '섹스', '자위', '가슴', '자위', '출장', '만남',
 '보지', '조건', '만남', '손흥민', '거미', '수지'], ['덕배', '때문', '맨시티', '손
 흥민', '맨시티'], ['어제', '개미', '대부분', '커뮤', '게시판', '초토화', '부들',
 '거리', '윤석열', '자살골', '이쯤', '손흥민', '푸스카스상'], ['프리미어리그',
 '타임', '베스트', '윙어', '투표', '최종', '후보', '손흥민', '베일', '포함'], ['프
 리미어리그', '타임', '베스트', '윙어', '투표', '최종', '후보', '손흥민', '베일',
 '포함'], [], ['나라', '문화', '만리장성', '우리나라', '제발', '헛소리', '손흥민',
 '단체', '대가리', '된거', '그날', '유랑'], ['어제', '개미', '대부분', '커뮤', '게
 시판', '초토화', '부들', '거리', '윤석열', '자살골', '이쯤', '손흥민', '푸스카스
 상'], ['여자', '신음', '판매', '섹스', '자위', '가슴', '자위', '출장', '만남', '보
 지', '크리스마스', '조건', '만남', '출장', '만남', '출장', '손흥민', '거미', '수
 지'], ['나라', '문화', '만리장성', '우리나라', '제발', '헛소리', '손흥민', '단
 체', '대가리', '된거', '그날', '유랑'], ['어제', '개미', '대부분', '커뮤', '게시

In [20]:

```
# 연관 분석을 수행
```

```
results = list(apriori(transactions,  
                      min_support=0.05,  
                      min_confidence=0.1,  
                      min_lift=5,  
                      max_length=2))  
print(results)
```

```
[RelationRecord(items=frozenset({'리트윗', '방법'}), support=0.069,  
ordered_statistics=[OrderedStatistic(items_base=frozenset({'리트  
윗'}), items_add=frozenset({'방법'}), confidence=1.0, lift=14.49275  
3623188404), OrderedStatistic(items_base=frozenset({'방법'}), ite  
ms_add=frozenset({'리트윗'}), confidence=1.0, lift=14.4927536231  
88404)]), RelationRecord(items=frozenset({'리트윗', '본인'}), suppo  
rt=0.067, ordered_statistics=[OrderedStatistic(items_base=frozen  
set({'리트윗'}), items_add=frozenset({'본인'}), confidence=0.971014  
4927536232, lift=14.279624893435633), OrderedStatistic(items_b  
ase=frozenset({'본인'}), items_add=frozenset({'리트윗'}), confidenc  
e=0.9852941176470588, lift=14.279624893435633)]), RelationRec  
ord(items=frozenset({'리트윗', '선수'}), support=0.069, ordered_sta  
tistics=[OrderedStatistic(items_base=frozenset({'리트윗'}), items_  
add=frozenset({'선수'}), confidence=1.0, lift=5.988023952095808),  
OrderedStatistic(items_base=frozenset({'선수'}), items_add=froze  
nset({'리트윗'}), confidence=0.4131736526946108, lift=5.98802395
```

In [21]:

```

# 데이터 프레임 형태로 정리
columns = ['source', 'target', 'support']
network_df = pd.DataFrame(columns=columns)

# 규칙의 조건절을 source, 결과절을 target, 지지도를 support 라는 데이터 프레임의 피처로
for result in results:
    if len(result.items) == 2:
        items = [x for x in result.items]
        row = [items[0], items[1], result.support]
        series = pd.Series(row, index=network_df.columns)
        network_df = network_df.append(series, ignore_index=True)

network_df.head()

```

Out[21]:

	source	target	support
0	리트윗	방법	0.069
1	리트윗	본인	0.067
2	리트윗	선수	0.069
3	리트윗	유튜브	0.069
4	리트윗	이벤트	0.069

[단어 빈도 추출하기]

말뭉치 추출

In [22]:

```
# 말뭉치를 추출함
tweet_corpus = "".join(df['ko_text'].tolist())
print(tweet_corpus)
```

손흥민이 떠오르며 기대할 수 밖에 없긴 한데 아쉬움도 좀 있어서 그러니까 느낌적으로 말하면 손흥민은 쉽지 않은 경기에서는 답을 내는 것 같은데 어려운 경기에서는 답 내는 모습을 본 적이 있나 없나 모르겠드아

여자신음 판매 섹스 자위가슴 자위영삼 출장만남 보지 크리스마스조건만남출장만남출장샵손흥민거미수지 홀쉬 그 손흥민 그 귀여운 사진 좀 나눔 해 주실분 ㅌㅌ 귀여운 사진 아무거나 관촬습니다토틀넘의 손흥민이 잉글랜드 프리미어리그 부상에서 복귀했다 아델 부르는 얀 반응 손흥민 손흥민 호날두베컴과 동급 최고 잉어 톱 선정 출장업소가슴 경주 대구 마사 마산 부산 소통 야동 인천 일탈 최신영화 손흥민 만남앱 소개팅 출장업소가슴 경주 대구 마사 마산 부산 소통 야동 인천 일탈 최신영화 손흥민 만남앱 소개팅 스포츠 팬은 두 종류 프로 구단의 팬 스타 선수의 팬롯데팬 손흥민팬부산 태생은 자연스레 롯데팬이 된다 나는 손흥민 때문에 토트넘팬손흥민이 맨시티로 이적하면 당연히 나는 맨시티팬 니들은 작은 나라 문화 흠쳐서 뭐하려고 만리장성도 우리나라 꺼라고 해볼까 제발 헛소리좀 그만했으면 좋겠다 정 떨어질때로 다 떨어지고 하다하다 손흥민까지 건들고 단체로 대가리가 어떻게 된거니 그날도 유랑 잘못 보내고출장업소가슴 경주 대구 마사 마산 부산 소통 야동 인천 일탈 최신영

In [23]:

```

from konlpy.tag import Okt
from collections import Counter

# 명사 키워드를 추출, 빈도수 계산
nouns_tagger = Okt()
nouns = nouns_tagger.nouns(tweet_corpus)
count = Counter(nouns)

# 한글자 키워드를 제거
remove_char_counter = Counter({x : count[x] for x in count if len(x) > 1})
print(remove_char_counter)

```

```

Counter({'손흥민': 1002, '설날': 327, '토틀넘': 280, '선수': 172, '참여': 144, '이벤트': 142, '새해': 117, '월드컵': 113, '만남': 88, '호랑이': 87, '축하': 80, '진출': 78, '이번': 71, '유튜브': 71, '포스트': 70, '팔로우': 70, '방법': 69, '리트윗': 69, '본인': 69, '채널': 68, '출장': 67, '친구': 67, '메시지': 59, '여러분': 53, '스포츠': 50, '모두': 49, '축구': 48, '당첨': 47, '명절': 46, '황희': 45, '준비': 45, '선물': 44, '가슴': 41, '문화': 38, '나라': 37, '만리장성': 37, '우리나라': 37, '제발': 37, '헛소리': 37, '단체': 37, '대가리': 37, '된거': 37, '그날': 37, '유랑': 37, '내일': 36, '벌써': 36, '복귀': 35, '카타르': 35, '조건': 33, '부산': 33, '연휴': 33, '자위': 32, '진짜': 31, '유니폼': 31, '그동안': 31, '싸인': 30, '현재': 27, '최신영': 26, '훈련': 26, '진행중': 26, '친필': 26, '타래': 26, '경주': 25, '대구': 25, '마사': 25, '마산': 25, '소통': 25, '야동': 25, '인천': 25, '소개팅': 25, '부상': 22, '윙어': 22, '오늘': 20, '출처': 20, '네이버': 19, '선행': 18, '확정': 18, '판매': 17, '맨시티': 17, '다음': 17, '연속': 17, '경기': 16, '여자': 16, '신음': 16, '섹스': 16, '보지': 16, '거미': 16, '롯데': 16, '최고': 15, '본선': 15, '이적': 14, '코스': 14, '어제': 14, '개미': 14, '대부분': 14, '커뮤': 14, '게시판': 14, '초토화': 14, '부들': 14, '거리': 14, '윤석열': 14, '자살골': 14, '이쯤': 14, '푸스카스상': 14, '뉴스': 14, '알리': 14, '수지': 13, '때문': 13, '후보': 13, '베일': 13, '영상': 13, '사진': 12, '선정': 12, '박지성': 12, '베스트': 12, '최종': 12, '캡틴': 12, '기자': 12, '매체': 12, '한국': 12, '프리미어리그': 11, '포함': 11, '눈물': 11, '대한민국': 11, '영국': 11, '설특집': 11, '구단': 10, '타임': 10, '투표': 10, '두리': 10, '완전': 10, '역대': 10, '취미': 10, '보기': 10, '트레이닝': 10, '세션': 10, '종류': 9, '스타': 9, '스포': 9, '티비': 9, '모든': 9, '감독': 9, '호날두': 8, '프로': 8, '태생': 8, '오징어': 8, '회화': 8, '곡성': 8, '엄마': 8, '아빠': 8, '리아': 8, '야생': 8, '여행': 8, '기록': 8, '리야': 8, '옹이': 8, '애인': 8, '대행': 8, '마사지': 8, '기흥': 8, '드림파크': 8, '야구장': 8, '소연': 8, '박우진':

```

8, '공유': 8, '설악산': 8, '국립공원': 8, '한플방뎡': 8, '역삼': 8, '원정': 8, '브라질': 8, '러시아': 8, '자기': 7, '구인': 7, '한국인': 7, '어깨': 7, '해외': 7, '애기': 7, '보고': 7, '정말': 7, '인사': 7, '시리아': 7, '케인': 7, '시장': 7, '조기': 7, '베컴': 6, '응원': 6, '나이': 6, '사이트': 6, '대화': 6, '합천군': 6, '부안': 6, '패피': 6, '아나고': 6, '발사': 6, '모임': 6, '추억': 6, '진유': 6, '광서': 6, '카메라': 6, '유럽': 6, '레전드': 6, '소식': 6, '형제': 6, '이강인': 6, '지금': 6, '시즌': 6, '투호': 6, '아들': 6, '확인': 6, '겨울': 6, '스트리밍': 6, '축구선수': 6, '근황': 6, '모습': 5, '아델': 5, '반응': 5, '꼭꼭': 5, '꾸벅': 5, '깍스': 5, '시간': 5, '통해': 5, '생각': 5, '소집': 5, '레바논': 5, '전망': 5, '에이스': 5, '수비': 5, '경우': 5, '기적': 5, '절친': 5, '대표팀': 5, '인터뷰': 5, '주장': 5, '연승': 5, '플랜': 5, '완성': 5, '연전': 5, '국가대표팀': 5, '애기': 5, '크리스마스': 4, '델리': 4, '환상': 4, '스포츠동아': 4, '이상': 4, '미소': 4, '나우': 4, '콘테': 4, '바로': 4, '추효주': 4, '스포츠조선': 4, '브로맨스': 4, '영미': 4, '리그': 4, '벤티지': 4, '불구': 4, '레저': 4, '노래': 4, '연합뉴스': 4, '권창훈': 4, '인스타': 4, '세계': 4, '처음': 4, '웃기': 4, '정도': 4, '흥민': 4, '머리': 4, '학원': 4, '케이': 4, '트윗': 4, '예선': 4, '이란': 4, '남편': 4, '정몽규': 4, '전달': 4, '세뱃돈': 4, '설명': 4, '느낌': 3, '거나': 3, '이름': 3, '홀랜드': 3, '진심': 3, '이유': 3, '마무리': 3, '풋볼': 3, '리스트': 3, '포스터': 3, '공격수': 3, '공개': 3, '중계': 3, '수도': 3, '당신': 3, '순간': 3, '계폭': 3, '기분': 3, '중동': 3, '황의조': 3, '황인': 3, '민아': 3, '완료': 3, '태극전사': 3, '목소리': 3, '동료': 3, '항상': 3, '생일': 3, '연봉': 3, '재벌': 3, '탄생': 3, '그냥': 3, '스토리': 3, '계속': 3, '따숨다': 3, '국가대표': 3, '스티커': 3, '이승우': 3, '이제': 3, '게다가': 3, '계약': 3, '영건': 3, '무력무력': 3, '러쉬': 3, '슈페': 3, '한번': 3, '서로': 3, '은퇴': 3, '제대로': 3, '동안': 3, '주말': 3, '게임': 3, '블랙': 3, '중국': 3, '판단력': 3, '순발력': 3, '슛팅': 3, '좌우': 3, '어디': 3, '서도': 3, '우리': 3, '선생님': 3, '퀴카': 3, '자체': 3, '지성': 3, '이형': 3, '계좌': 3, '손찬박': 3, '랜선': 3, '약간': 3, '코스프레': 2, '설득': 2, '발표': 2, '전쟁': 2, '거임': 2, '년차': 2, '차로': 2, '부장': 2, '기수': 2, '센터': 2, '개멋': 2, '핵심': 2, '일한국': 2, '애칭': 2, '맑은': 2, '최애': 2, '출전': 2, '스파이더맨': 2, '봉준호': 2, '영화': 2, '연예': 2, '현장': 2, '여성': 2, '외침': 2, '좌절': 2, '포포투': 2, '노컷뉴스': 2, '출격': 2, '발행': 2, '일시': 2, '오후': 2, '최신': 2, '안경': 2, '선배': 2, '전차': 2, '군단': 2, '몰락': 2, '기억': 2, '막내': 2, '기만': 2, '소니': 2, '이즈': 2, '화일': 2, '강서': 2, '플러스': 2, '엔트리': 2, '몸싸움': 2, '다각': 2, '일괄': 2, '우대': 2, '배경': 2, '배송': 2, '반택': 2, '편택': 2, '디엠': 2, '침대': 2, '밀집': 2, '완벽': 2, '파훼': 2, '비결': 2, '카타르월드컵': 2, '경범': 2, '회복': 2, '예상': 2, '평양': 2, '코로나': 2, '환희': 2, '작별': 2, '전력': 2, '보강': 2, '브라이턴': 2, '지역': 2, '쿠키뉴스': 2, '사이': 2, '로메': 2, '리터': 2, '은근': 2, '허당': 2, '버스': 2, '빠빠': 2, '거의': 2, '코리아': 2, '외국인': 2, '사람': 2, '얼굴': 2, '전혀': 2, '레버쿠젠': 2, '요즘': 2, '바둑': 2, '전과': 2, '막산': 2, '막말': 2, '거짓말': 2, '아치': 2, '조폭': 2, '집중': 2, '방탄': 2, '레이디': 2, '히아신스':

2, '타이밍': 2, '웰케': 2, '가발': 2, '열애': 2, '뭔가': 2, '탈락': 2, '이재성': 2, '국내': 2, '언론': 2, '조명': 2, '조유민': 2, '스키': 2, '동경': 2, '정상': 2, '송민규': 2, '이동준': 2, '김진수': 2, '울버햄프턴': 2, '통산': 2, '자리': 2, '멀리': 2, '나마': 2, '기쁨': 2, '개인': 2, '후반': 2, '결승골': 2, '가지': 2, '점수': 2, '눈치': 2, '자력': 2, '고생': 2, '미니': 2, '안심': 2, '기용': 2, '회사': 2, '광주': 2, '사고': 2, '꼰대': 2, '없슴': 2, '김민재': 2, '폴핏': 2, '후반기': 2, '시작': 2, '임영웅': 2, '개그맨': 2, '브랜드': 2, '평판': 2, '인구': 2, '법도': 2, '데돈': 2, '투자': 2, '누가': 2, '세유클로버': 2, '설연휴': 2, '포스': 2, '청와대': 2, '사랑': 2, '실력': 2, '에브라': 2, '라운드': 2, '안팎': 2, '케미': 2, '다시': 2, '코리안리거': 2, '역시': 2, '현생': 2, '중고': 2, '뉴비': 2, '라떼': 2, '토비': 2, '위주': 2, '흔적': 2, '바이오': 2, '지원': 2, '해설': 2, '가의': 2, '질타': 2, '네이트뉴스': 2, '특집': 2, '맷돌': 2, '두부': 2, '치기': 2, '어의': 2, '음식': 2, '먹방': 2, '벤덱': 2, '고스톱': 2, '직접': 2, '제기차기': 2, '떡국': 2, '만들기': 2, '리스': 2, '웃놀이': 2, '해파리': 2, '리기': 2, '아쉬움': 1, '홀쉬': 1, '나눔': 1, '습니': 1, '잉글랜드': 1, '하면황런쥘': 1, '마크': 1, '제노': 1, '재민': 1, '수녀': 1, '귀신': 1, '스페인어': 1, '일단': 1, '덕배': 1, '참으로': 1, '관성': 1, '오피셜': 1, '저녁': 1, '문명': 1, '알겠어': 1, '승리': 1, '김치': 1, '갈비찜': 1, '비비고': 1, '김연아': 1, '싸이': 1, '강남스타일': 1, '도리': 1, '포즈': 1, '앙큼깜찍': 1, '스킨': 1, '하나': 1, '췌트': 1, '솔로': 1, '출연': 1, '직업': 1, '정리': 1, '결혼': 1, '커플': 1, '블로그': 1, '무료': 1, '메이저리그': 1, '통신원': 1, '수첩': 1, '좀비': 1, '사태': 1, '말함': 1, '축가': 1, '중이': 1, '잠시': 1, '아시아': 1, '빅리그': 1, '어시스트': 1, '먹뚱': 1, '차범근': 1, '국뽕': 1, '크립': 1, '토러스': 1, '랜드': 1, '오픈': 1, '심상정': 1, '정의당': 1, '지지율': 1, '우레이': 1, '성적': 1, '요새': 1, '볼때': 1, '런가': 1, '웨이': 1, '애가': 1, '조합': 1, '대신': 1, '만약': 1, '인종차별': 1, '흑인': 1, '데이': 1, '이하': 1, '생략': 1, '저장': 1, '고추': 1, '효주': 1, '토크': 1, '선진': 1, '출국': 1, '최대': 1, '발견': 1, '투톱': 1, '가능': 1, '스발': 1, '애기야': 1, '토론회': 1, '부디': 1, '악재': 1, '기회': 1, '약점': 1, '지적': 1, '전술': 1, '실험': 1, '얼마만큼': 1, '정확성': 1, '세밀': 1, '초점': 1, '당시': 1, '도하': 1, '명구': 1, '하의': 1, '이준석': 1, '피시방': 1, '연습생': 1, '평가': 1, '구렛나루': 1, '밀고': 1, '조원희': 1, '하니': 1, '종합': 1, '카페': 1, '아프리카': 1, '가차': 1, '부분': 1, '원래': 1, '꼬꼬리': 1, '빠빠야': 1, '절규': 1, '뚫이': 1, '자격': 1, '노력': 1, '보상': 1, '어찌': 1, '워커': 1, '우정': 1, '육사': 1, '침투': 1, '주저': 1, '수가': 1, '동작': 1, '장난': 1, '전문': 1, '미디어': 1, '주신': 1, '온라인': 1, '세리머니': 1, '십오야': 1, '기념': 1, '셀카': 1, '한장': 1, '진액': 1, '멸균': 1, '거즈': 1, '시크': 1, '전준홍': 1, '허브': 1, '라벤더': 1, '잉크': 1, '구매': 1, '기숙': 1, '스텐드': 1, '시계': 1, '줄무늬': 1, '가디건': 1, '자동차': 1, '홍송': 1, '가격': 1, '남매': 1, '정치': 1, '태권도': 1, '정육점': 1, '알바': 1, '축구화': 1, '유나': 1, '위시': 1, '여서': 1, '누구': 1, '자신': 1, '마음': 1, '파트너': 1, '인터': 1, '거가': 1, '여기': 1, '먼저': 1, '슈미트': 1, '대박': 1, '그대': 1, '대사': 1, '누군가': 1, '마치': 1, '존버': 1, '몰

입': 1, '멤버': 1, '사인': 1, '붕붕': 1, '필폰': 1, '남작': 1, '이후': 1, '몬티': 1, '모차르트': 1, '울동': 1, '소리': 1, '듣기': 1, '소개': 1, '정작': 1, '먹구름': 1, '화분': 1, '적응': 1, '울음': 1, '무대': 1, '풍선': 1, '대충': 1, '대사도': 1, '포인트': 1, '손아섭': 1, '박성진': 1, '특필': 1, '팩트': 1, '보도': 1, '콘텐츠': 1, '코리안': 1, '시청': 1, '검색': 1, '알고리즘': 1, '울브스': 1, '인도': 1, '명의': 1, '긴장': 1, '매력': 1, '여유': 1, '로움': 1, '자신감': 1, '나물': 1, '겨우': 1, '정우영': 1, '용이': 1, '대안': 1, '체력': 1, '볼컨': 1, '로안': 1, '매번': 1, '지지': 1, '증명사진': 1, '김정민': 1, '증사': 1, '찬증사': 1, '성적표': 1, '루셈': 1, '벤탄쿠르': 1, '이준': 1, '명과': 1, '예비': 1, '포워드': 1, '백승호': 1, '남태희': 1, '김진규': 1, '디렉': 1, '고등학교': 1, '청춘': 1, '신문부': 1, '소속': 1, '교실': 1, '키스': 1, '비즈니스석': 1, '타고': 1, '가라': 1, '에스프레소': 1, '오라': 1, '아무': 1, '목적': 1, '직관': 1, '상상': 1, '훗날': 1, '인생': 1, '즐거움': 1, '차지': 1, '구체': 1, '걸스데이': 1, '입술': 1, '바짝': 1, '회연': 1, '국민': 1, '진골': 1, '이도': 1, '남녀': 1, '동반': 1, '심지어': 1, '전반': 1, '컨디션': 1, '아보': 1, '이대영': 1, '지난': 1, '사실': 1, '결정': 1, '지은': 1, '굳이': 1, '그때': 1, '아마': 1, '조가': 1, '만큼': 1, '가치': 1, '문제': 1, '역량': 1, '구멍': 1, '역할': 1, '빈도': 1, '갑자기': 1, '이크': 1, '기네': 1, '와중': 1, '주심': 1, '뒷모습': 1, '빈자리': 1, '핑크': 1, '유재석': 1, '한참': 1, '프로레슬링': 1, '일본': 1, '스즈키': 1, '노루': 1, '시나': 1, '고요': 1, '야구': 1, '서건창': 1, '이정후': 1, '용병': 1, '교육': 1, '번은': 1, '저번': 1, '우영': 1, '다나': 1, '왕세자': 1, '다년': 1, '정부': 1, '베이징': 1, '올림픽': 1, '입성': 1, '화이팅': 1, '스마트': 1, '외치': 1, '마스코트': 1, '호돌이': 1, '전호': 1, '줄라': 1, '성남': 1, '이재명': 1, '임기': 1, '평균': 1, '작년': 1, '원금': 1, '억원': 1, '구장': 1, '신축': 1, '윤경': 1, '쿠션': 1, '후원': 1, '대한': 1, '무게': 1, '절대': 1, '구단주': 1, '회장': 1, '멍청이': 1, '만도': 1, '인성': 1, '멘탈': 1, '리더쉽': 1, '존나': 1, '전방': 1, '해축': 1, '트친소': 1, '이자': 1, '절찬': 1, '다른': 1, '가게': 1, '정이': 1, '건가': 1, '다그': 1, '래야': 1, '분데스리가': 1, '골대': 1, '불운': 1, '무승부': 1, '방출': 1, '델레': 1, '에버턴': 1, '입단': 1, '소희': 1, '대관': 1, '현철': 1, '광수': 1, '시원': 1, '태진아': 1, '하드': 1, '강찬희': 1, '한복': 1, '체험': 1, '세배': 1, '미션': 1, '소매': 1, '라인': 1, '레쉬': 1, '가드': 1, '어쩔리': 1, '찰떡': 1, '우주': 1, '화이': 1, '티익': 1, '큐트': 1, '딸바보': 1, '이신': 1, '돌파': 1, '감염': 1, '운동선수': 1, '달라': 1, '니엄': 1, '마임': 1, '비질': 1, '바코드': 1, '개좇': 1, '씨발': 1, '삭제': 1, '재업': 1, '스트레스': 1, '총알': 1, '상황': 1, '호스': 1, '신고': 1, '위안': 1, '상대방': 1, '태클': 1})

단어 빈도 점수 추가

In [24]:

```
# 키워드와 키워드 빈도 점수를 'node', 'nodesize' 라는 데이터 프레임의 피처로 생성
node_df = pd.DataFrame(remove_char_counter.items(), columns=['node', 'nodesize'])
node_df = node_df[node_df['nodesize'] >= 50] # 시각화의 편의를 위해 'nodesize' 50 이하
node_df.head()
```

Out[24]:

	node	nodesize
0	손흥민	1002
11	출장	67
12	만남	88
23	토틀넘	280
45	스포츠	50

<Step4. 시각화> : 연관 키워드 네트워크 시각화

[연관 키워드 네트워크 시각화]

- 아래 코드 실행을 위해, anaconda prompt 혹은 Terminal에서 아래와 같은 패키지를 설치해 줍니다.
 - (env_name) pip install networkx
- 혹은 아래의 코드로 라이브러리를 설치합니다.
- networkx 버전 2.3 이하를 설치하는 경우, 아래 코드를 다음과 같이 바꿔주어야 합니다.
 - sizes = [G.nodes[node]['nodesize']*25 for node in G] -> sizes = [G.node[node]['nodesize']*25 for node in G]

In [25]:

```
!pip install networkx
```

Requirement already satisfied: networkx in c:\users\yj\anaconda3\lib\site-packages (2.6.3)

In [26]:

```
import networkx as nx
plt.figure(figsize=(25,25))

# networkx 그래프 객체를 생성
G = nx.Graph()

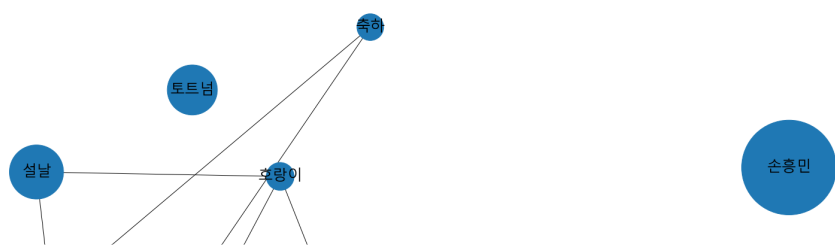
# node_df의 키워드 빈도수를 데이터로 하여, 네트워크 그래프의 '노드' 역할을 하는 원을 생성
for index, row in node_df.iterrows():
    G.add_node(row['node'], nodesize=row['nodesize'])

# network_df의 연관 분석 데이터를 기반으로, 네트워크 그래프의 '관계' 역할을 하는 선을 생성
for index, row in network_df.iterrows():
    G.add_weighted_edges_from([(row['source'], row['target'], row['support'])])

# 그래프 디자인과 관련된 파라미터를 설정
pos = nx.spring_layout(G, k=0.6, iterations=50)
# nx.draw(G, pos=pos)
sizes = [G.nodes[node]['nodesize']*25 for node in G]
nx.draw(G, pos=pos, node_size=sizes)

# Windows 사용자는 AppleGothic 대신, 'Malgun Gothic'. 그 외 OS는 OS에서 한글을 지원
# nx.draw_networkx_labels(G, pos=pos, font_family='NanumGothic', font_size=25)
nx.draw_networkx_labels(G, pos=pos, font_family='Malgun Gothic', font_size=25)

# 그래프를 출력
ax = plt.gca()
plt.show()
```



In []: