1. kiwipiepy 설치

-> 터미널에 pip install kiwipiepy==0.7.6 입력

2. pyvis 설치

-> 터미널에 pip install pyvis 입력

4. 아래 코드 복사

import re # 정규표현식 패키지  
import tomotopy as tp # 토픽 모델링에 사용할 패키지  
from kiwipiepy import Kiwi #형태소 분석에 사용할 패키지  
from pyvis.network import Network # 네트워크 시각화에 사용할 패키지  
  
try:  
 # 이미 전처리된 코퍼스가 있으면 불러온다.  
 corpus = tp.utils.Corpus.load('k.cps')  
except IOError:  
 # 없으면 전처리를 시작한다.  
 kiwi = Kiwi()  
 kiwi.prepare()  
  
 # 형태소 분석 후 사용할 태그  
 pat\_tag = re.compile('NN[GP]|V[VA]|MAG|MM')  
  
  
 def tokenizer(raw, user\_data):  
 res, \_ = user\_data()[0]  
 for w, tag, start, l in res:  
 if not pat\_tag.match(tag) or len(w) <= 1: continue  
 yield w + ('다' if tag.startswith('V') else ''), start, l  
  
  
 corpus = tp.utils.Corpus(  
 tokenizer=tokenizer  
 )  
 # 입력 파일에는 한 줄에 문헌 하나씩이 들어가 있습니다.  
 corpus.process((line, kiwi.async\_analyze(line)) for line in open('/Users/xiu0327/lab/감성분석/data/38766\_output.txt', encoding='utf-8'))  
 # 전처리한 코퍼스를 저장한다.  
 corpus.save('k.cps')  
  
# 최소 10개 이상 문헌에 등장하고, 전체 출현빈도는 20 이상인 단어만 사용합니다.  
# 그리고 상위 10개 고빈도 단어는 분석에서 제외하구요  
# 주제 개수는 40개입니다.  
mdl = tp.CTModel(tw=tp.TermWeight.ONE, min\_df=10, min\_cf=20, rm\_top=10, k=40, corpus=corpus)  
mdl.train(0)  
  
# 문헌 수가 만 개 이상이라면 num\_beta\_sample을 1~5 정도로 줄여도 됨  
# 수 천개라면 최소 10 정도, 수 백개에 불과하다면 20 이상으로 키우는걸 권장  
mdl.num\_beta\_sample = 1  
print('Num docs:{}, Num Vocabs:{}, Total Words:{}'.format(  
 len(mdl.docs), len(mdl.used\_vocabs), mdl.num\_words  
))  
print('Removed Top words: ', \*mdl.removed\_top\_words)  
  
# 총 1000회 깁스샘플링을 반복합니다.  
# 적절한 반복회수는 데이터에 따라 다릅니다.  
# ll\_per\_word 값의 증가가 둔화되거나 멈추는 지점까지만  
# 반복하는걸 추천합니다.  
for i in range(0, 1000, 20):  
 print('Iteration: {:04}, LL per word: {:.4}'.format(i, mdl.ll\_per\_word))  
 mdl.train(20)  
print('Iteration: {:04}, LL per word: {:.4}'.format(i, mdl.ll\_per\_word))  
  
# 학습된 결과를 시각화 합니다.  
g = Network(width=800, height=800, font\_color="#333")  
correl = mdl.get\_correlations().reshape([-1])  
correl.sort()  
  
# 상관계수 상위 10%만 간선으로 잇습니다.  
top\_tenth = mdl.k \* (mdl.k - 1) // 10  
top\_tenth = correl[-mdl.k - top\_tenth]  
  
topic\_counts = mdl.get\_count\_by\_topics()  
  
for k in range(mdl.k):  
 label = "#{}".format(k)  
 title = ' '.join(word for word, \_ in mdl.get\_topic\_words(k, top\_n=8))  
 print('Topic', label, title)  
 label += '\n' + ' '.join(word for word, \_ in mdl.get\_topic\_words(k, top\_n=3))  
 g.add\_node(k, label=label, title=title, shape='ellipse', value=float(topic\_counts[k]))  
 for l, correlation in zip(range(k - 1), mdl.get\_correlations(k)):  
 if correlation < top\_tenth: continue  
 g.add\_edge(k, l, value=float(correlation), title='{:.02}'.format(correlation))  
  
g.barnes\_hut(gravity=-1000, spring\_length=20)  
g.show\_buttons()  
# 시각화 파일이 topic\_network.html 이라는 이름으로 저장됩니다.  
# 웹 브라우저로 열어서 확인해보세요.  
g.show("topic\_network.html")