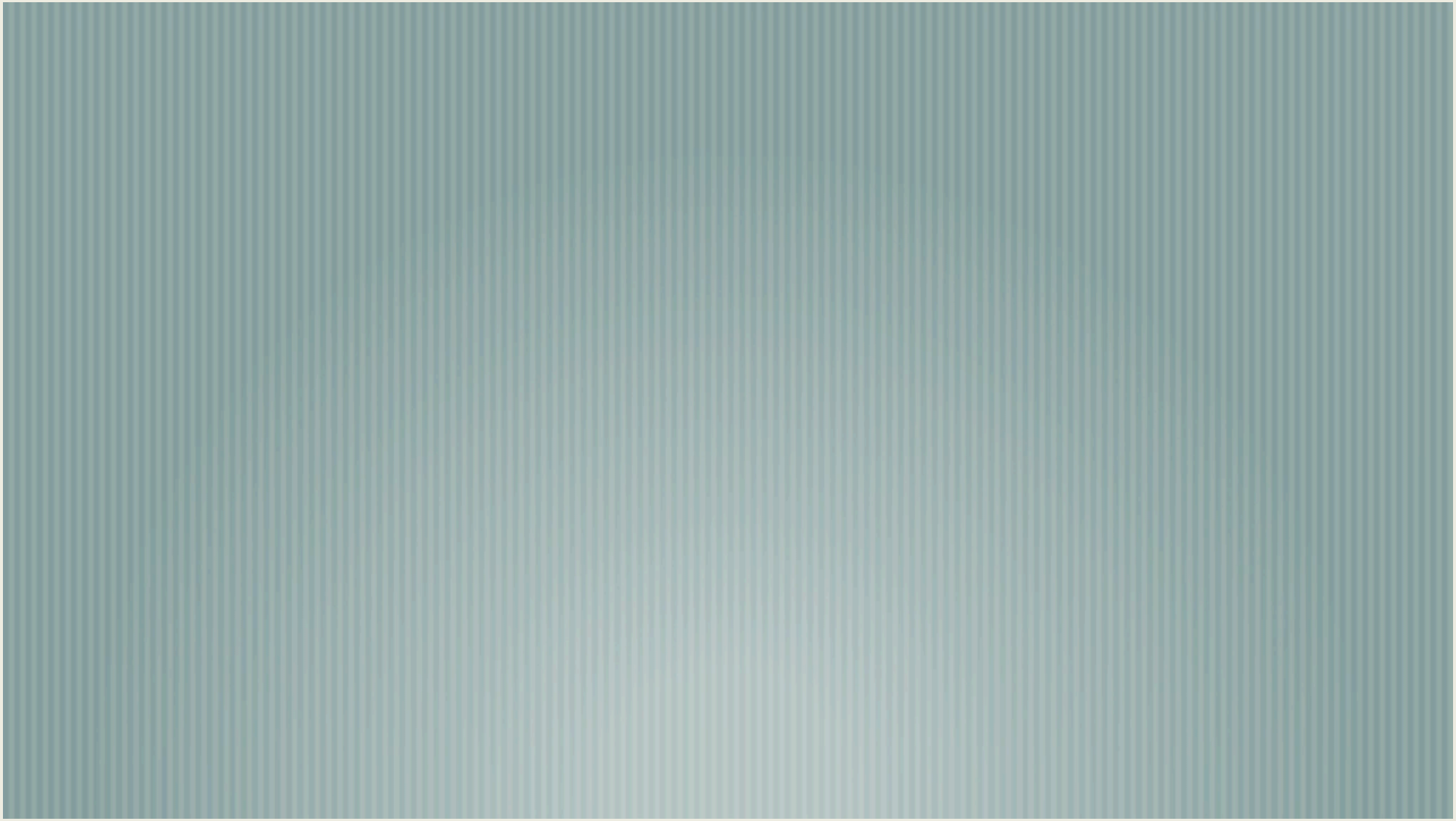
Topic modeling and Visualization

Korea University, DAVIAN Lab.

강경필

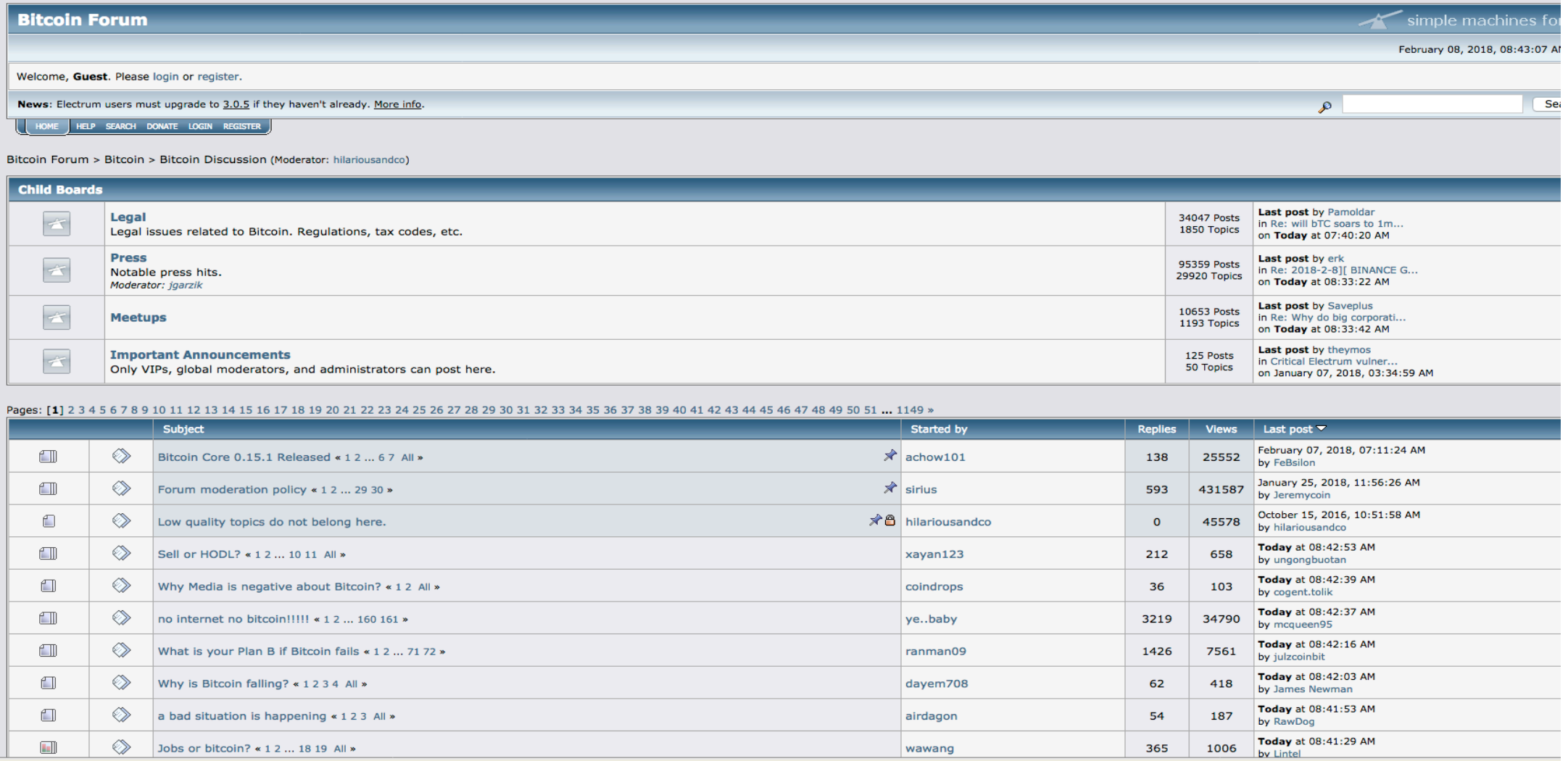
Topic modeling & Visualization



Bitcoin


트윗
님이 리트윗했습니다
안녕
"""""비트코인 샀어야 했는데"
라고 생각 했을 때 샀어야 했는데"
라고 생각 했을 때 샀어야 했는데"
라고 생각 했을 때 샀어야 했는데
2017. 10. 13. 22:59

Bitcoin Forum data



Bitcoin Forum simple machines fo

February 08, 2018, 08:43:07 A

Welcome, **Guest**. Please login or register.

News: Electrum users must upgrade to 3.0.5 if they haven't already. More info.

HOME | HELP | SEARCH | DONATE | LOGIN | REGISTER

Bitcoin Forum > Bitcoin > Bitcoin Discussion (Moderator: hilariousandco)

Child Boards

Legal Legal issues related to Bitcoin. Regulations, tax codes, etc.	34047 Posts 1850 Topics	**Last post** by Pamoldar in Re: will bTC soars to 1m... on **Today** at 07:40:20 AM
Press Notable press hits. *Moderator: jgarzik*	95359 Posts 29920 Topics	**Last post** by erk in Re: 2018-2-8][BINANCE G... on **Today** at 08:33:22 AM
Meetups	10653 Posts 1193 Topics	**Last post** by Saveplus in Re: Why do big corporati... on **Today** at 08:33:42 AM
Important Announcements Only VIPs, global moderators, and administrators can post here.	125 Posts 50 Topics	**Last post** by theymos in Critical Electrum vulner... on January 07, 2018, 03:34:59 AM

Pages: [**1**] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 **...** 1149 »

Subject	Started by	Replies	Views	Last post
Bitcoin Core 0.15.1 Released « 1 2 ... 6 7 All »	achow101	138	25552	February 07, 2018, 07:11:24 AM by FeBsilon
Forum moderation policy « 1 2 ... 29 30 »	sirius	593	431587	January 25, 2018, 11:56:26 AM by Jeremycoin
Low quality topics do not belong here.	hilariousandco	0	45578	October 15, 2016, 10:51:58 AM by hilariousandco
Sell or HODL? « 1 2 ... 10 11 All »	xayan123	212	658	**Today** at 08:42:53 AM by ungongbuotan
Why Media is negative about Bitcoin? « 1 2 All »	coindrops	36	103	**Today** at 08:42:39 AM by cogent.tolik
no internet no bitcoin!!!!! « 1 2 ... 160 161 »	ye..baby	3219	34790	**Today** at 08:42:37 AM by mcqueen95
What is your Plan B if Bitcoin fails « 1 2 ... 71 72 »	ranman09	1426	7561	**Today** at 08:42:16 AM by julzcoinbit
Why is Bitcoin falling? « 1 2 3 4 All »	dayem708	62	418	**Today** at 08:42:03 AM by James Newman
a bad situation is happening « 1 2 3 All »	airdagon	54	187	**Today** at 08:41:53 AM by RawDog
Jobs or bitcoin? « 1 2 ... 18 19 All »	wawang	365	1006	**Today** at 08:41:29 AM by Lintel

Bitcoin Forum data

Author	Topic: Should I Wait or Invest On Bitcoin? (Read 12069 times)

mamataindia
Newbie

Activity: 1
Merit: 0

Should I Wait or Invest On Bitcoin?
October 25, 2017, 06:35:06 PM

#1

I am newbie on bitcoin, recently introduced with the community. I have seen a lot of positive feedback about bitcoin investment. I am not interested do business or trade on market. I want to used it for long term investment only. Some people suggested me anytime is good for invest on bitcoin but I want to do it wisely. If you have some idea of the market and analysis kindly do it for me.

MissionPhailed
Hero Member

Activity: 560
Merit: 502

Re: Should I Wait or Invest On Bitcoin?
October 25, 2017, 06:41:17 PM

#2

Well, value of Bitcoin has risen quite significantly in the last six months and chances are a correction (drop in value) due to people selling off their profits is around the corner. You cold sort of spread the risk by buying an amount of BTC each week or month or any time interval you like. If the price drops and you continue to purchase BTC, the *average entry value* of your BTC drops also. If values go up you'll already go in the green numbers.

hacker1001101001
Full Member

Activity: 182
Merit: 100

Re: Should I Wait or Invest On Bitcoin?
October 25, 2017, 06:44:14 PM

#3

Waiting is not need now to invest in bitcoin and people keep on investing in bitcoin and fain profit. If peopel invest in bitcoin now people will get profit in comming future as bitcoin is accepted to reach heights in prices in comming future.

Bitcoin Forum data

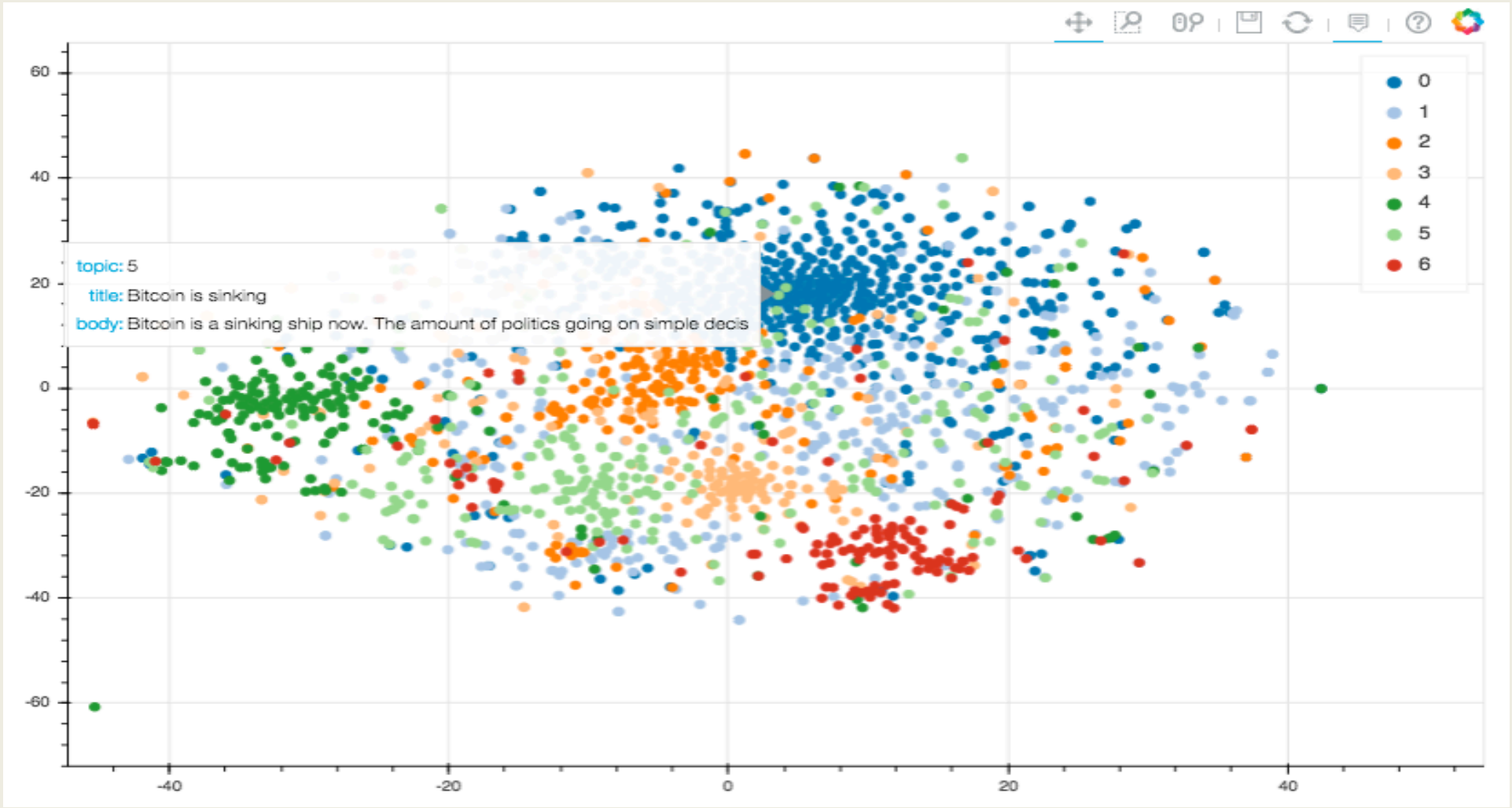
```
print(len(posts))
print(posts[0])
```

executed in 7ms, finished 19:31:06 2018-02-08

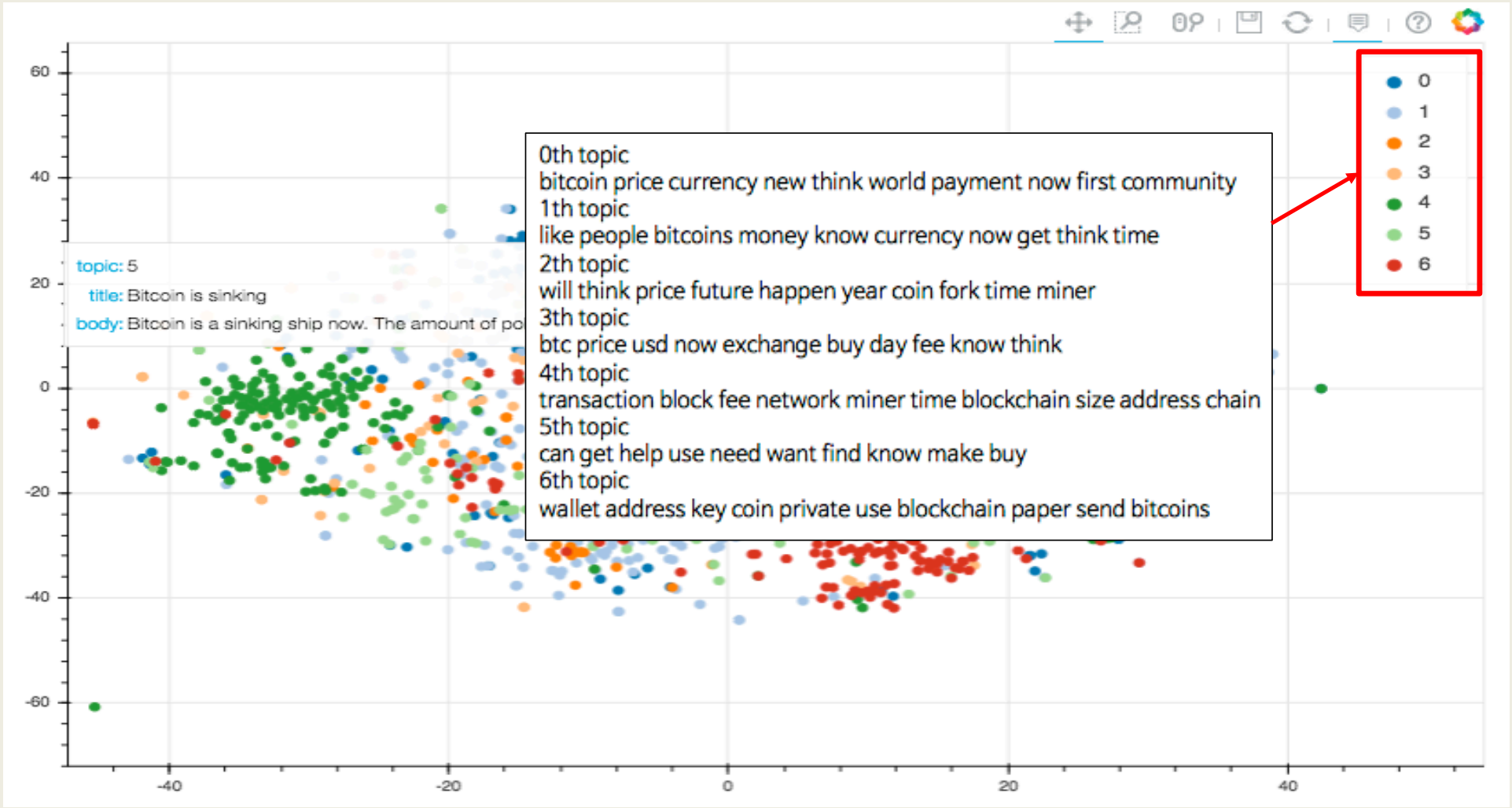
40641

{'title': "HELP I'm a Bitcoin Dummy!", 'time': 'December 12, 2017, 10:40:17 PM', 'body': 'I am new to Bitcoin, but recently invested in a small amount to explore how it works. I₩'ve ran over the basics, and will be doing my own research, but it₩'s always valuable to ask a community for more down-to-earth explanations of things, so here I am. If you have credible and verifiable information, I would be delighted if you could contribute to this learning process, for myself and other people able to view this thread. If you choose to respond, please cite the specific question you are answering with a designation. (example: "Q1" For multiples, please include the designation before each separate response.)Thank you, in advance, to anyone that participates. Please answer the questions, while assuming the following:I have a Bitcoin WalletI have NOT lost any information and have full access to all aspects of said Wallet.I have Bitcoin in my Wallet, account, etc.Q1: Can I secure my Bitcoin on paper? Is there a way that I can write down certain information ON PAPER and store that information in a safe, file cabinet, safety deposit box, etc? If so, can I then SAFELY delete the digital Wallet on my device without losing access to the account and Bitcoins? Q2: What information about a Wallet IS and ISN₩'T safe to share with other people? Q3: What are the different ways a person can convert Bitcoin to U.S. Dollars (USD), and how does it work? Q4: Is it taxed (either as Bitcoin or once converted to USD), and if so, how is it taxed (form type?), what is it taxed as (income, interest, etc.)? BONUS QUESTION:Q5: How do I sell or give Bitcoin to another person? Let₩'s say it₩'s Christmas, a birthday, special occasion or whatever and I want to gift Bitcoin to someone. How do I do that? Or, on a regular day, how do I sell Bitcoin to a regular person, in real regular life? Thanks again, in advance, to anyone that can contribute. I hope this helps me, and anyone else interested. '}

What we will implement…



What we will implement…

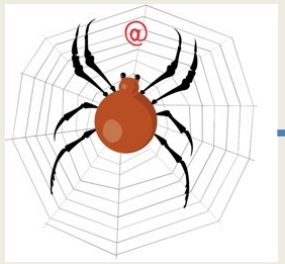


Before we start…

- Anaconda python 3.6
- Library
  - pip install ujson (만약 설치 안되면 넘어가도 괜찮습니다.)
  - pip install stop_words
  - python 실행 후
    - import nltk
    - nltk.download() 실행 후 wordnet 설치

- Code & dataset:
  - https://github.com/rudvlf0413/bitcoin_topicmodeling
  - * 크롤링 코드
  - https://github.com/rudvlf0413/crawler/tree/master/bitcoin_forum

Workflow



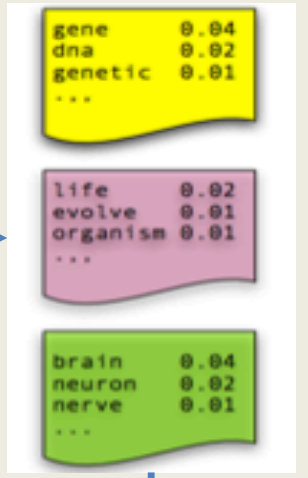
Crawling
웹페이지에서 데이터 수집

Data preprocessing
parsing,
stemming, lemmatizing
stop-words 제거
word indexing
…

Term document matrix 구축

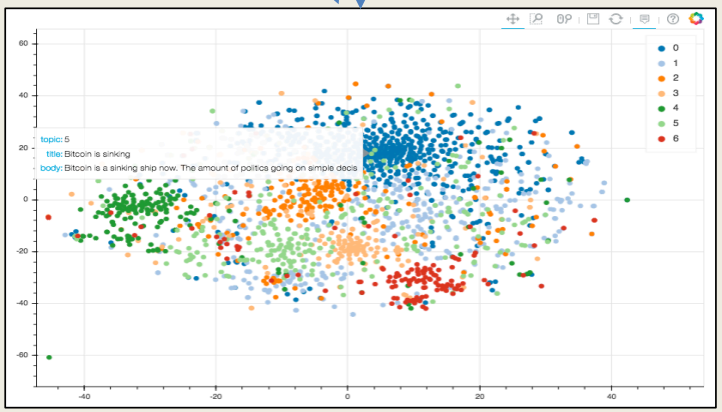
	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Topic modeling (NMF)



t-SNE를 통해
저차원 벡터 계산

topic keywords 추출



Visualization

Preprocessing

```
stopwords = set(stop_words.get_stop_words('en'))
stopwords.update(['quote', 'pmquote', 'amquote', 'just', 'don', 'one', 'thing', 'even', 'way', 'maybe', 'also', 'please', 'well', 'actually', 'something',
                  'going', 'anything', 'le', 'ever', 'say', 'see', 'likely', 'per', 'another', 'someone', 'let', 'anyone', 'doesn', 'include', 'doe'])
lemmatizer = WordNetLemmatizer()
```

xecuted in 391ms, finished 14:55:14 2018-02-08

```
def parse_string(input_string):
    input_string = input_string.lower()
    input_string = re.sub(r'http₩S+', ' ', input_string)
    words = re.sub("[^a-zA-Z]", " ", input_string).split()
    words = [lemmatizer.lemmatize(w) for w in words]
    words = [w for w in words if w not in stopwords and len(w) > 2]
    return words
```

의미 없는 단어 제거

```
post_words = parse_string(post['body'])
post_words = [w for w in post_words if word_freq[w] >= 10]
if len(post_words) < 5:
    continue
```

저빈도 단어 제거

매우 짧은 길이의 문장 제거

Preprocessing

```
stopwords = set(stop_words.get_stop_words('en'))
stopwords.update(['quote', 'pmquote', 'amquote', 'just', 'don', 'one', 'thing', 'even', 'way', 'maybe', 'also', 'please', 'well', 'actually', 'something',
                  'going', 'anything', 'le', 'ever', 'say', 'see', 'likely', 'per', 'another', 'someone', 'let', 'anyone', 'doesn', 'include', 'doe'])
lemmatizer = WordNetLemmatizer()
```

xecuted in 391ms, finished 14:55:14 2018-02-08

대문자를 소문자로 변경

본문에서 url 제거

특수기호 제거(regular expression)

```
def parse_string(input_string):
    input_string = input_string.lower()
    input_string = re.sub(r'http₩S+', ' ', input_string)
    words = re.sub("[^a-zA-Z]", " ", input_string).split()
    words = [lemmatizer.lemmatize(w) for w in words]
    words = [w for w in words if w not in stopwords and len(w) > 2]
    return words
```

```
post_words = parse_string(post['body'])
post_words = [w for w in post_words if word_freq[w] >= 10]
if len(post_words) < 5:
    continue
```

Preprocessing

```
stopwords = set(stop_words.get_stop_words('en'))
stopwords.update(['quote', 'pmquote', 'amquote', 'just', 'don', 'one', 'thing', 'even', 'way', 'maybe', 'also', 'please', 'well', 'actually', 'something',
                  'going', 'anything', 'le', 'ever', 'say', 'see', 'likely', 'per', 'another', 'someone', 'let', 'anyone', 'doesn', 'include', 'doe'])
lemmatizer = WordNetLemmatizer()
```

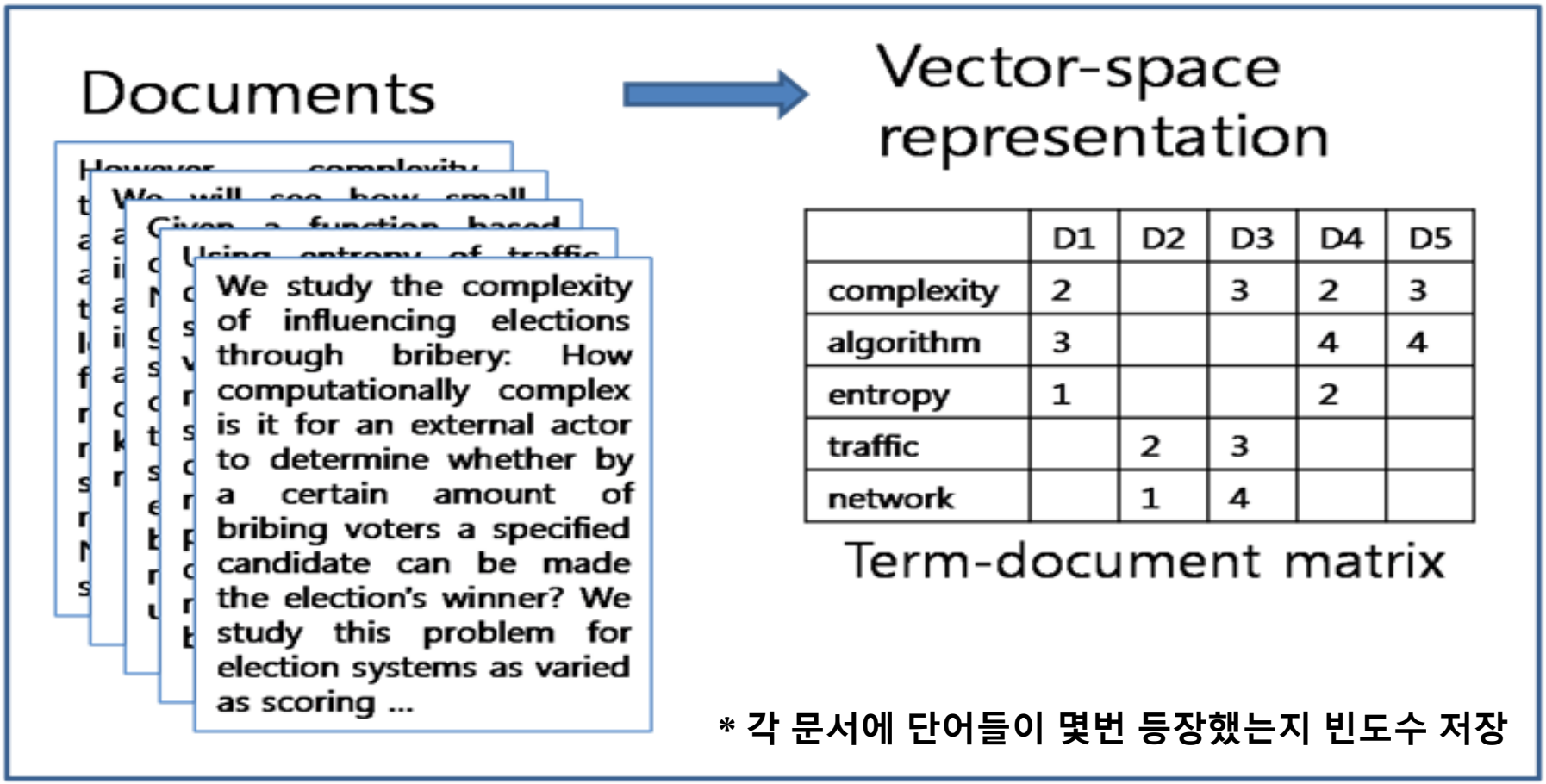
xecuted in 391ms, finished 14:55:14 2018-02-08

```
def parse_string(input_string):
    input_string = input_string.lower()
    input_string = re.sub(r'http₩S+', ' ', input_string)
    words = re.sub("[^a-zA-Z]", " ", input_string).split()
    words = [lemmatizer.lemmatize(w) for w in words]
    words = [w for w in words if w not in stopwords and len(w) > 2]
    return words
```

lemmatize; 복수를 단수로 수정

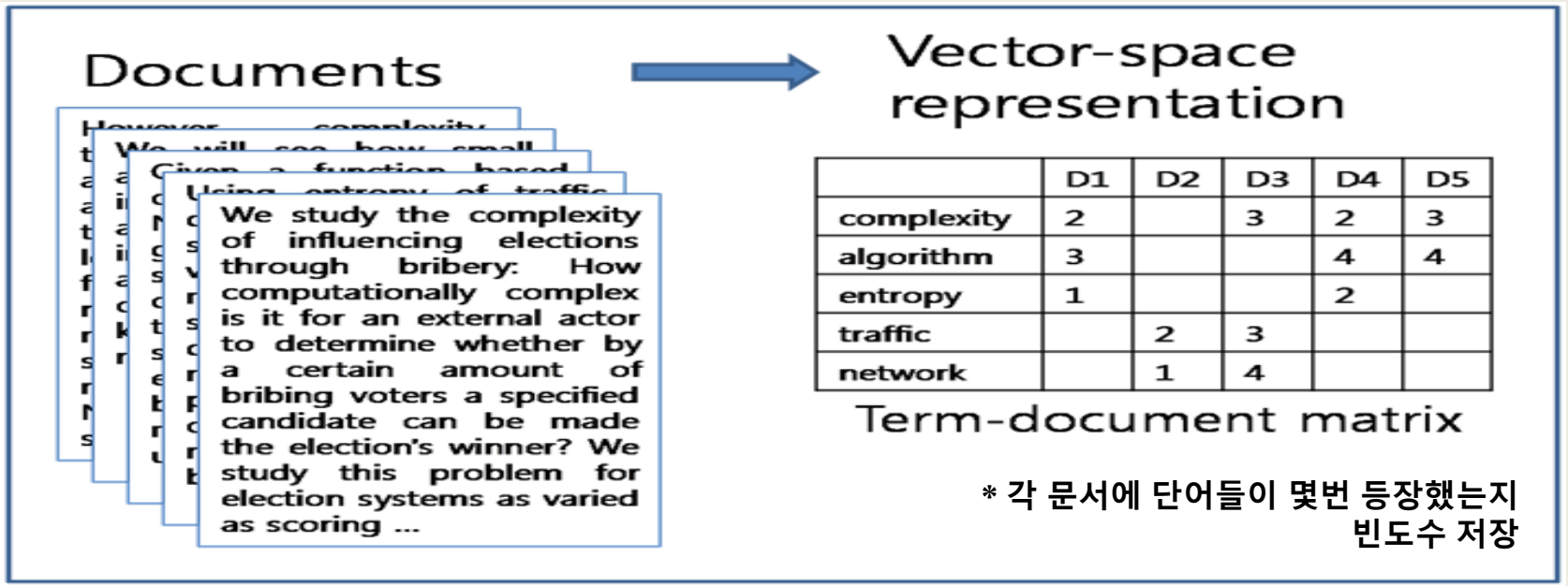
```
post_words = parse_string(post['body'])
post_words = [w for w in post_words if word_freq[w] >= 10]
if len(post_words) < 5:
    continue
```

Term document matrix



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term document matrix



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

* 각 문서에 단어들이 몇번 등장했는지 빈도수 저장

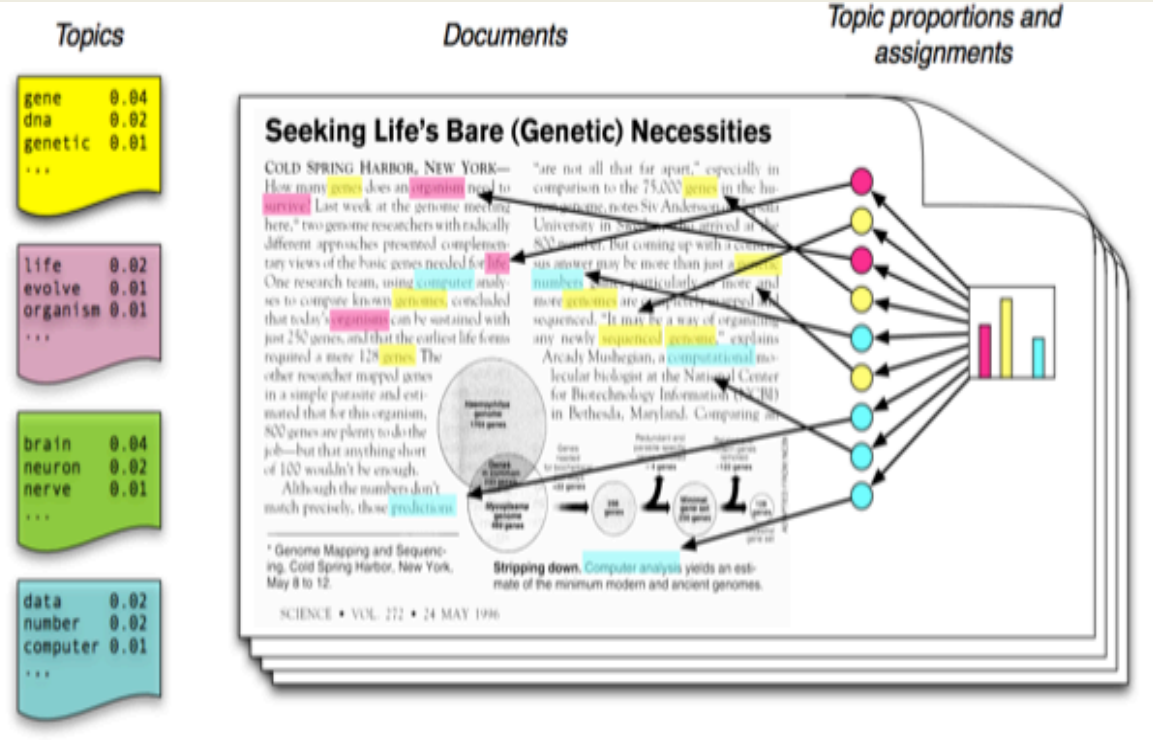
```
tdm = dok_matrix((len(preprocessed_data), len(voca)), dtype=np.float32)
for i, post in enumerate(preprocessed_data):
  for word in post['words']:
    tdm[i, word2id[word]] += 1

tdm = tdm.tocsr()
tdm = normalize(tdm)
```

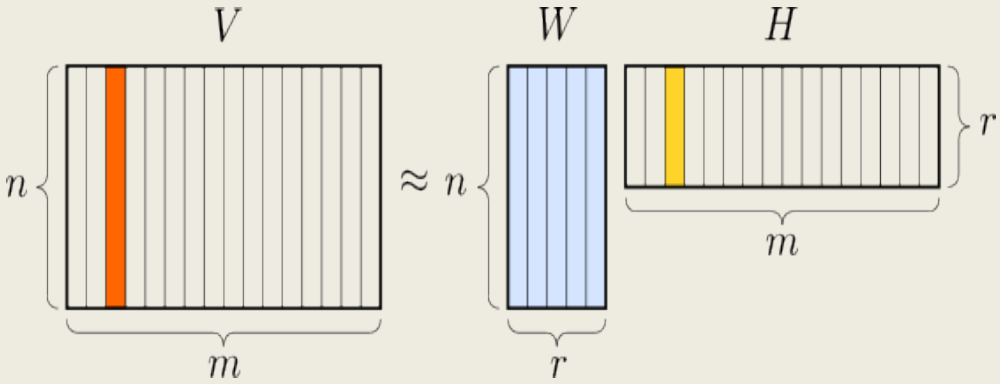
Sparse matrix 생성

빈도수 저장

$$||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$ 각 행의 길이를 1로 normalize

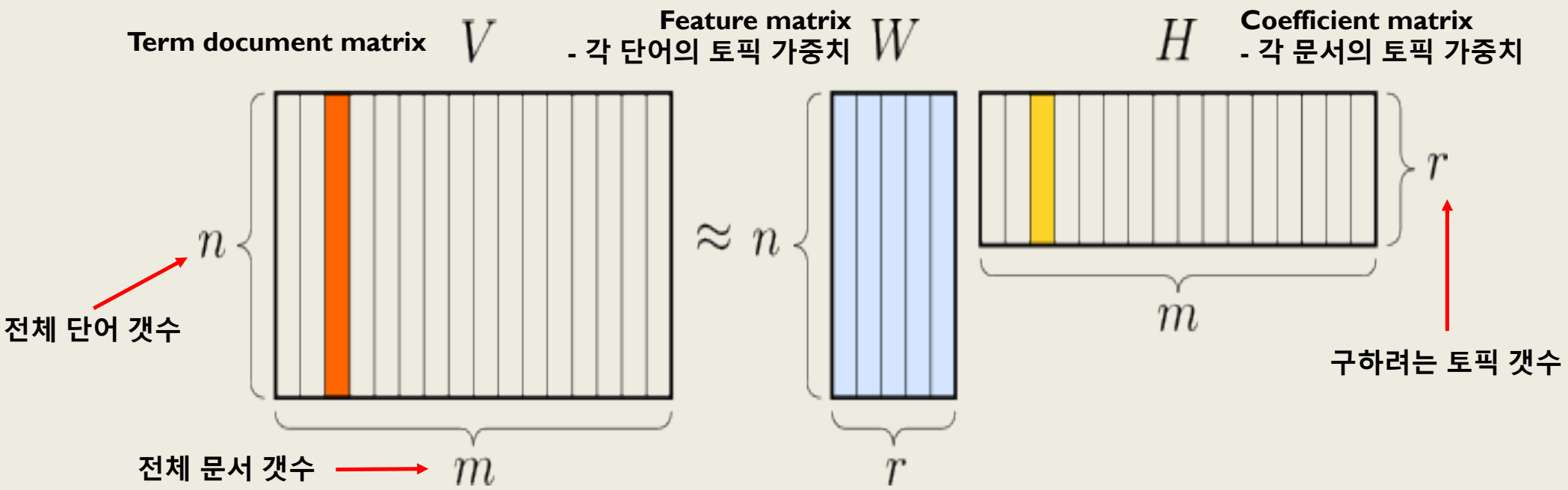
Topic modeling





Nonnegative matrix factorization

Topic modeling - NMF



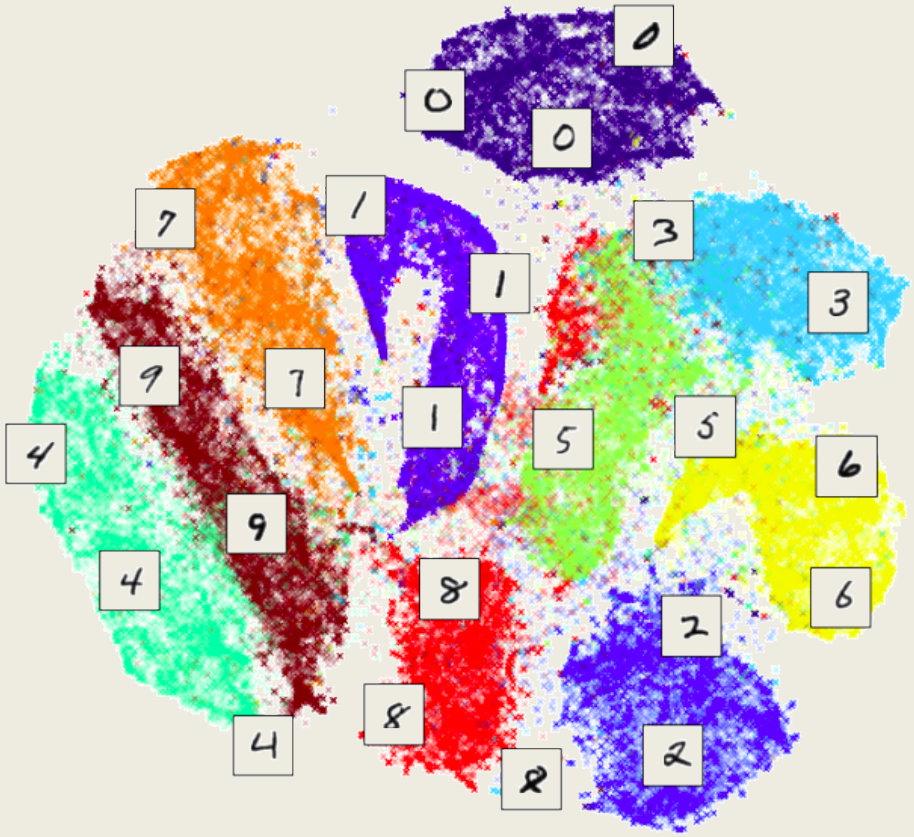
$$\min_{W,H} ||V - WH||_F, subject\ to\ W \geq 0, H \geq 0$$

Visualization

t-SNE (t-distributed Stochastic Neighbor Embedding)

- 고차원의 데이터를 효과적으로 저차원으로 차원축소하는 알고리즘

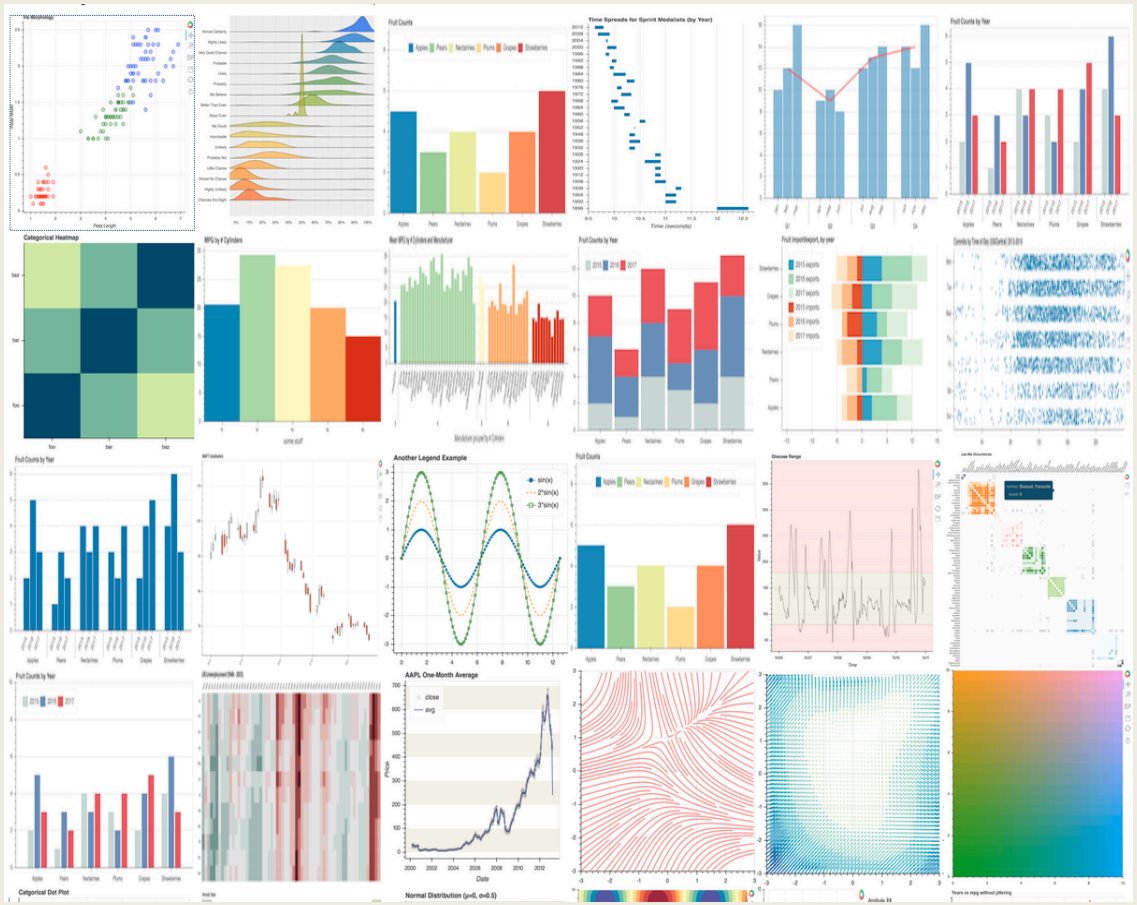
1. 원래 공간(N차원)에서 각 거리를 계산
2. 원래 공간에서의 거리에 따른 확률 계산(p_{ij})
3. M차원에서의 각 데이터 위치를 랜덤하게 초기화
4. 마찬가지로 M차원에서의 거리를 계산하고
   거리에 따른 확률 계산 (q_{ij})
5. M차원에서의 확률분포가 N차원의서의 확률분포와
   가까워지도록 위치를 재조정



MNIST Dataset

0 ~9 숫자의 필기체 사진과 레이블으로 구성된 데이터셋

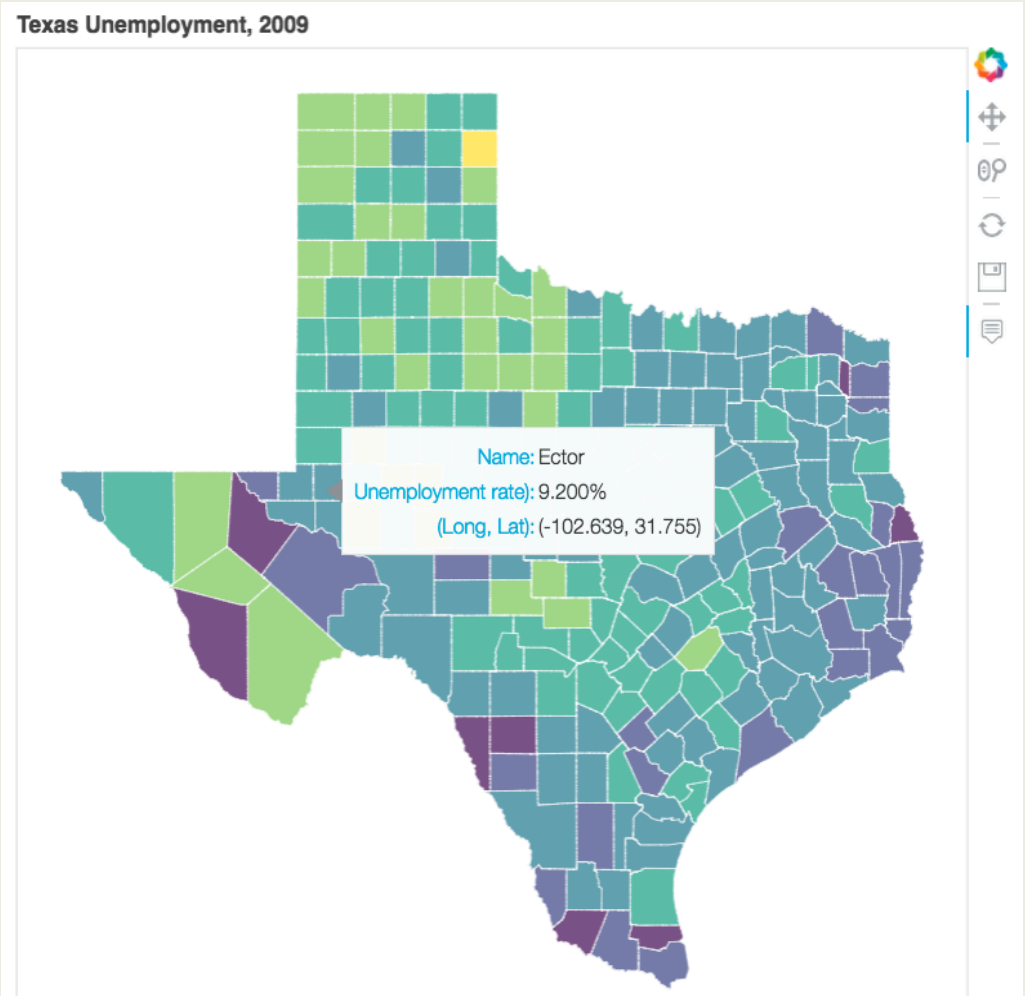
Visualization - Bokeh



Bokeh 시각화 예제

- 웹 기반의 시각화에 특화된 파이썬 라이브러리
- Matplotlib보다 쉽고 간결하게 시각화 가능
- 다양한 시각화 예제 및 모델 제공
- 다양한 상호작용 기능 지원
- html 코드로 export 가능
- jupyter notebook에서 바로 시각화 가능

Bokeh 예제

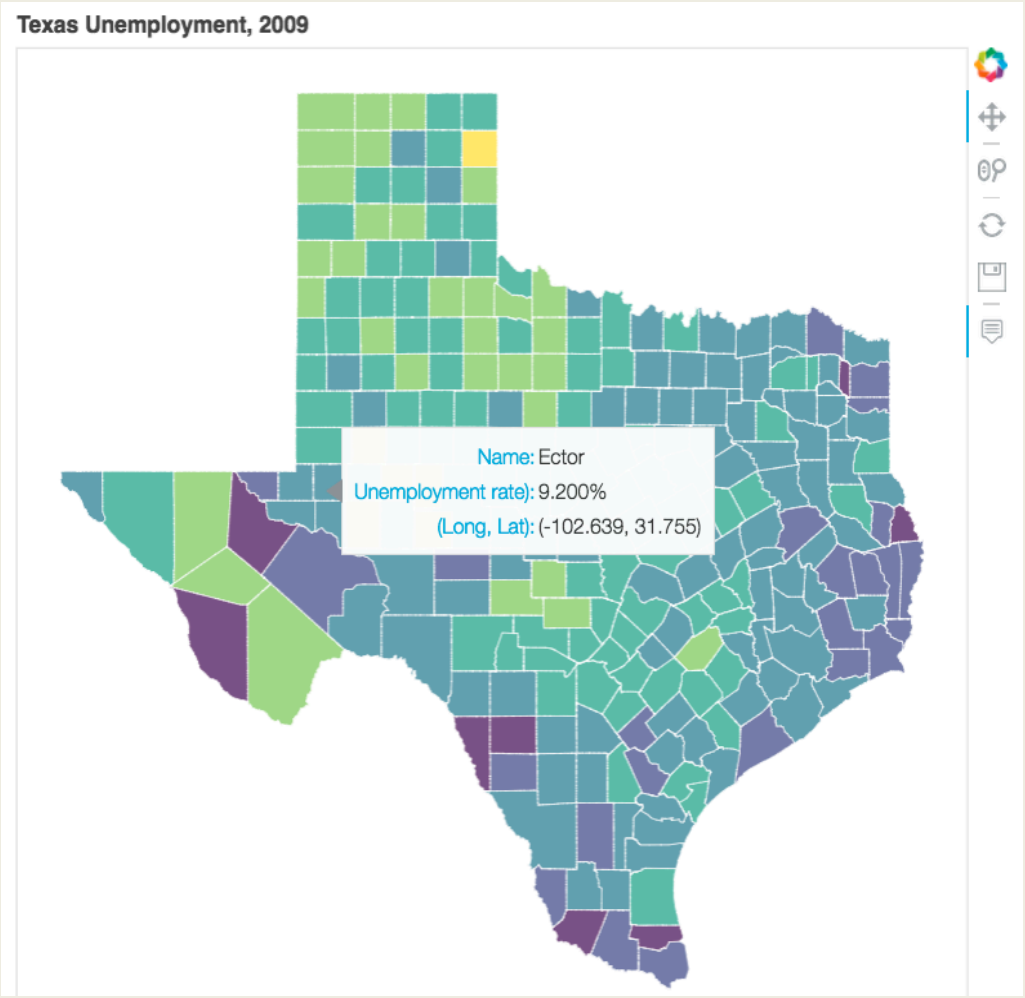


시각화에 사용할 데이터를 bokeh 포맷으로

캔버스 객체 생성 및 설정

```
source = ColumnDataSource(data=dict(
    x=county_xs,
    y=county_ys,
    name=county_names,
    rate=county_rates,
))

TOOLS = "pan,wheel_zoom,reset,hover,save"

p = figure(
    title="Texas Unemployment, 2009", tools=TOOLS,
    x_axis_location=None, y_axis_location=None
)
p.grid.grid_line_color = None

p.patches('x', 'y', source=source,
          fill_color={'field': 'rate', 'transform': color_mapper},
          fill_alpha=0.7, line_color="white", line_width=0.5)

hover = p.select_one(HoverTool)
hover.point_policy = "follow_mouse"
hover.tooltips = [
    ("Name", "@name"),
    ("Unemployment rate)", "@rate%"),
    ("(Long, Lat)", "($x, $y)"),
]

show(p)
```

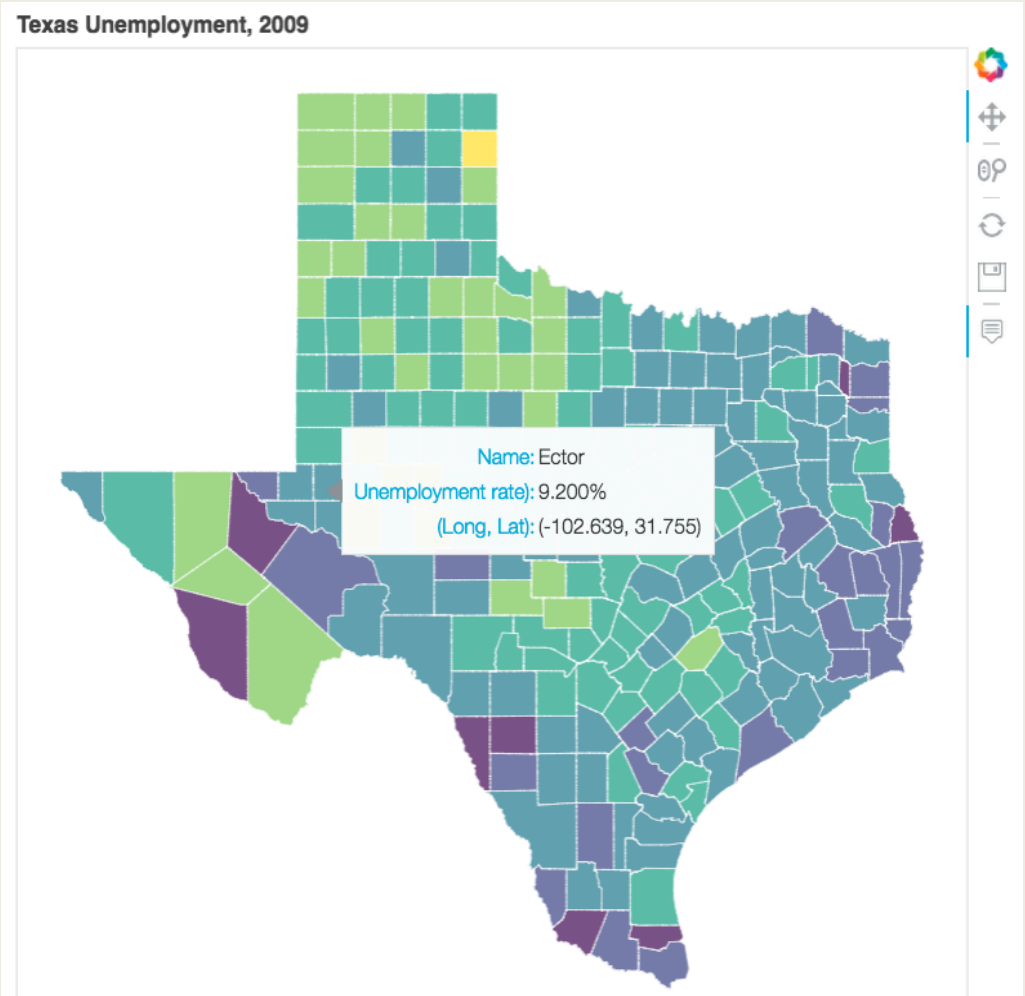
Bokeh 예제




```
source = ColumnDataSource(data=dict(
    x=county_xs,
    y=county_ys,
    name=county_names,
    rate=county_rates,
))

TOOLS = "pan,wheel_zoom,reset,hover,save"

p = figure(
    title="Texas Unemployment, 2009", tools=TOOLS,
    x_axis_location=None, y_axis_location=None
)
p.grid.grid_line_color = None

p.patches('x', 'y', source=source,
          fill_color={'field': 'rate', 'transform': color_mapper},
          fill_alpha=0.7, line_color="white", line_width=0.5)

hover = p.select_one(HoverTool)
hover.point_policy = "follow_mouse"
hover.tooltips = [
    ("Name", "@name"),
    ("Unemployment rate)", "@rate%"),
    ("(Long, Lat)", "($x, $y)"),
]

show(p)
```

각각 지도 패치 추가

Bokeh 예제

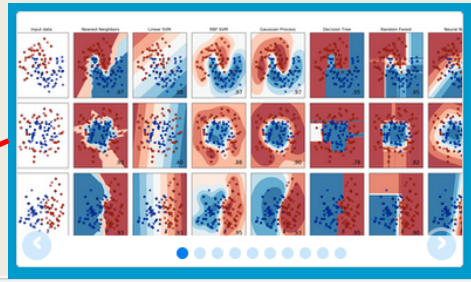



```
source = ColumnDataSource(data=dict(
    x=county_xs,
    y=county_ys,
    name=county_names,
    rate=county_rates,
))

TOOLS = "pan,wheel_zoom,reset,hover,save"

p = figure(
    title="Texas Unemployment, 2009", tools=TOOLS,
    x_axis_location=None, y_axis_location=None
)
p.grid.grid_line_color = None

p.patches('x', 'y', source=source,
          fill_color={'field': 'rate', 'transform': color_mapper},
          fill_alpha=0.7, line_color="white", line_width=0.5)

hover = p.select_one(HoverTool)
hover.point_policy = "follow_mouse"
hover.tooltips = [
    ("Name", "@name"),
    ("Unemployment rate)", "@rate%"),
    ("(Long, Lat)", "($x, $y)"),
]

show(p)
```

tooltip interaction 추가

화면에 보여주기

실습



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

```
from sklearn.decomposition import NMF
from sklearn.preprocessing import normalize
from collections import Counter
from scipy.sparse import dok_matrix
import numpy as np
from nltk.stem import WordNetLemmatizer
import stop_words
import ujson as json
import re

stopwords = set(stop_words.get_stop_words('en'))
stopwords.update(['quote', 'pmquote', 'amquote', 'just', 'don', 'one', 'thing', 'even', 'way', 'maybe', 'also', 'please', 'well', 'actually', 'something',
                  'going', 'anything', 'le', 'ever', 'say', 'see', 'likely', 'per', 'another', 'someone', 'let', 'anyone', 'doesn', 'include', 'doe'])
lemmatizer = WordNetLemmatizer()
```

필요한 라이브러리 import

stopword 리스트 및 lemmatizer 생성

```
with open('./bitcoin_post.json') as f:
    posts = json.load(f)
```

데이터 load

executed in 274ms, finished 19:31:03 2018-02-08

```
print(len(posts))
print(posts[0])
```

executed in 7ms, finished 19:31:06 2018-02-08

40641

{'title': "HELP I'm a Bitcoin Dummy!", 'time': 'December 12, 2017, 10:40:17 PM', 'body': 'I am new to Bitcoin, but recently invested in a small amount to explore how it works. I₩'ve ran over the basics, and will be doing my own research, but it₩'s always valuable to ask a community for more down-to-earth explanations of things, so here I am. If you have credible and verifiable information, I would be delighted if you could contribute to this learning process, for myself and other people able to view this thread. If you choose to respond, please cite the specific question you are answering with a designation. (example: "Q1" For multiples, please include the designation before each separate response.)Thank you, in advance, to anyone that participates. Please answer the questions, while assuming the following:I have a Bitcoin WalletI have NOT lost any information and have full access to all aspects of said Wallet.I have Bitcoin in my Wallet, account, etc.Q1: Can I secure my Bitcoin on paper? Is there a way that I can write down certain information ON PAPER and store that information in a safe, file cabinet, safety deposit box, etc? If so, can I then SAFELY delete the digital Wallet on my device without losing access to the account and Bitcoins? Q2: What information about a Wallet IS and ISN₩'T safe to share with other people? Q3: What are the different ways a person can convert Bitcoin to U.S. Dollars (USD), and how does it work? Q4: Is it taxed (either as Bitcoin or once converted to USD), and if so, how is it taxed (form type?), what is it taxed as (income, interest, etc.)? BONUS QUESTION:Q5: How do I sell or give Bitcoin to another person? Let₩'s say it₩'s Christmas, a birthday, special occasion or whatever and I want to gift Bitcoin to someone. How do I do that? Or, on a regular day, how do I sell Bitcoin to a regular person, in real regular life? Thanks again, in advance, to anyone that can contribute. I hope this helps me, and anyone else interested. '}

```
preprocessed_data = []
voca = set()
word_freq = Counter()

with open('./bitcoin_post.json') as f:
    posts = json.load(f)
    for post in posts:
        post_words = parse_string(post['body'])
        word_freq.update(post_words)

with open('./bitcoin_post.json') as f:
    posts = json.load(f)
    for i, post in enumerate(posts):
        post_words = parse_string(post['body'])
        post_words = [w for w in post_words if word_freq[w] >= 10]
        if len(post_words) < 5:
            continue

        voca.update(post_words)
        post['words'] = post_words
        preprocessed_data.append(post)

voca = list(voca)
word2id = {w: i for i, w in enumerate(voca)}
del posts
```

각 단어의 빈도수 계산

각 본문에 대해 전처리 후 단어열 생성

전체 단어에 대해 vocabulary 생성

단어 indexing

```
def parse_string(input_string):
    input_string = input_string.lower()
    input_string = re.sub(r'http₩S+', ' ', input_string)
    words = re.sub("[^a-zA-Z]", " ", input_string).split()
    words = [lemmatizer.lemmatize(w) for w in words]
    words = [w for w in words if w not in stopwords and len(w) > 2]
    return words
```

전처리 함수

```
tdm = dok_matrix((len(preprocessed_data), len(voca)), dtype=np.float32)
for i, post in enumerate(preprocessed_data):
    for word in post['words']:
        tdm[i, word2id[word]] += 1

tdm = tdm.tocsr()
tdm = normalize(tdm)
```

Term document matrix 생성

document vector 길이가 1로 normalize

```
K = 7
nmf = NMF(n_components=K)
W = nmf.fit_transform(tdm)
H = nmf.components_
```

NMF 계산

```
for k in range(K):
    print(f"{k}th topic")
    for idx in H[k].argsort()[::-1][:10]:
        print(voca[idx], end=' ')
    print()
```

k번째 토픽에 대해 weight가 높은 상위 10개의 단어 추출

executed in 35ms, finished 15:29:11 2018-02-08

```
0th topic
bitcoin price currency new think world payment now first community
1th topic
like people bitcoins money know currency now get think time
2th topic
will think price future happen year coin fork time miner
3th topic
btc price usd now exchange buy day fee know think
4th topic
transaction block fee network miner time blockchain size address chain
5th topic
can get help use need want find know make buy
6th topic
wallet address key coin private use blockchain paper send bitcoins
```

전체 데이터를 시각화하면 시간이 오래 걸리므로
2000개의 데이터만 랜덤하게 추출

```
from sklearn.manifold import TSNE

random_index = np.random.choice(len(preprocessed_data), size=2000)
document_2d = TSNE(init='pca').fit_transform(tdm[random_index].toarray())

document_topic = W[random_index, :].argmax(axis=1)
topic_document_indexes = [[] for i in range(K)]
for i, topic in enumerate(document_topic):
    topic_document_indexes[topic].append(i)
```

t-SNE를 통해
2차원 document vector 계산

각 document가 몇번째 토픽인지를 계산

```
from bokeh.models import HoverTool, Legend
from bokeh.palettes import Category20
from bokeh.io import show, output_notebook
from bokeh.plotting import figure, ColumnDataSource
output_notebook()

# 사용할 툴들
p = figure(plot_width=900, plot_height=600,
           toolbar_location='above', x_range=(document_2d[:, 0].min()*1.05, document_2d[:, 1].max()*1.2))

# 각 토픽별 그래프에 추가하도록 source data 생성
circles = []
for k, document_indexes in enumerate(topic_document_indexes):
  document_source = ColumnDataSource(data={
    'x': document_2d[document_indexes, 0],
    'y': document_2d[document_indexes, 1],
    'topic': [k for _ in document_indexes],
    'title': [preprocessed_data[random_index[i]]['title'] for i in document_indexes],
    'body': [preprocessed_data[random_index[i]]['body'][:75] for i in document_indexes],
    'color': [Category20[10][k] for _ in document_indexes],
  })
  circles.append(p.circle('x', 'y', color='color', legend='topic', source=document_source, size=6))

# 몇가지 interaction
p.add_tools(HoverTool(tooltips=[('topic', '@topic'), ("title", "@title"), ('body', '@body')], renderers=circles, mode='mouse'))
p.legend.click_policy = 'hide'
show(p)
```

캔버스 객체 생성

x축 범위 설정

각 토픽에 대해 bokeh 데이터 객체 정의

캔버스에 circle 생성

```
from bokeh.models import HoverTool, Legend
from bokeh.palettes import Category20
from bokeh.io import show, output_notebook
from bokeh.plotting import figure, ColumnDataSource
output_notebook()

# 사용할 툴들
p = figure(plot_width=900, plot_height=600,
           toolbar_location='above', x_range=(document_2d[:, 0].min()*1.05, document_2d[:, 1].max()*1.2))

# 각 토픽별 그래프에 추가하도록 source data 생성
circles = []
for k, document_indexes in enumerate(topic_document_indexes):
  document_source = ColumnDataSource(data={
    'x': document_2d[document_indexes, 0],
    'y': document_2d[document_indexes, 1],
    'topic': [k for _ in document_indexes],
    'title': [preprocessed_data[random_index[i]]['title'] for i in document_indexes],
    'body': [preprocessed_data[random_index[i]]['body'][:75] for i in document_indexes],
    'color': [Category20[10][k] for _ in document_indexes],
  })
  circles.append(p.circle('x', 'y', color='color', legend='topic', source=document_source, size=6))

# 몇가지 interaction
p.add_tools(HoverTool(tooltips=[('topic', '@topic'), ("title", "@title"), ('body', '@body')], renderers=circles, mode='mouse'))
p.legend.click_policy = 'hide'
show(p)
```

각 토픽에 대해 bokeh 데이터 객체 정의

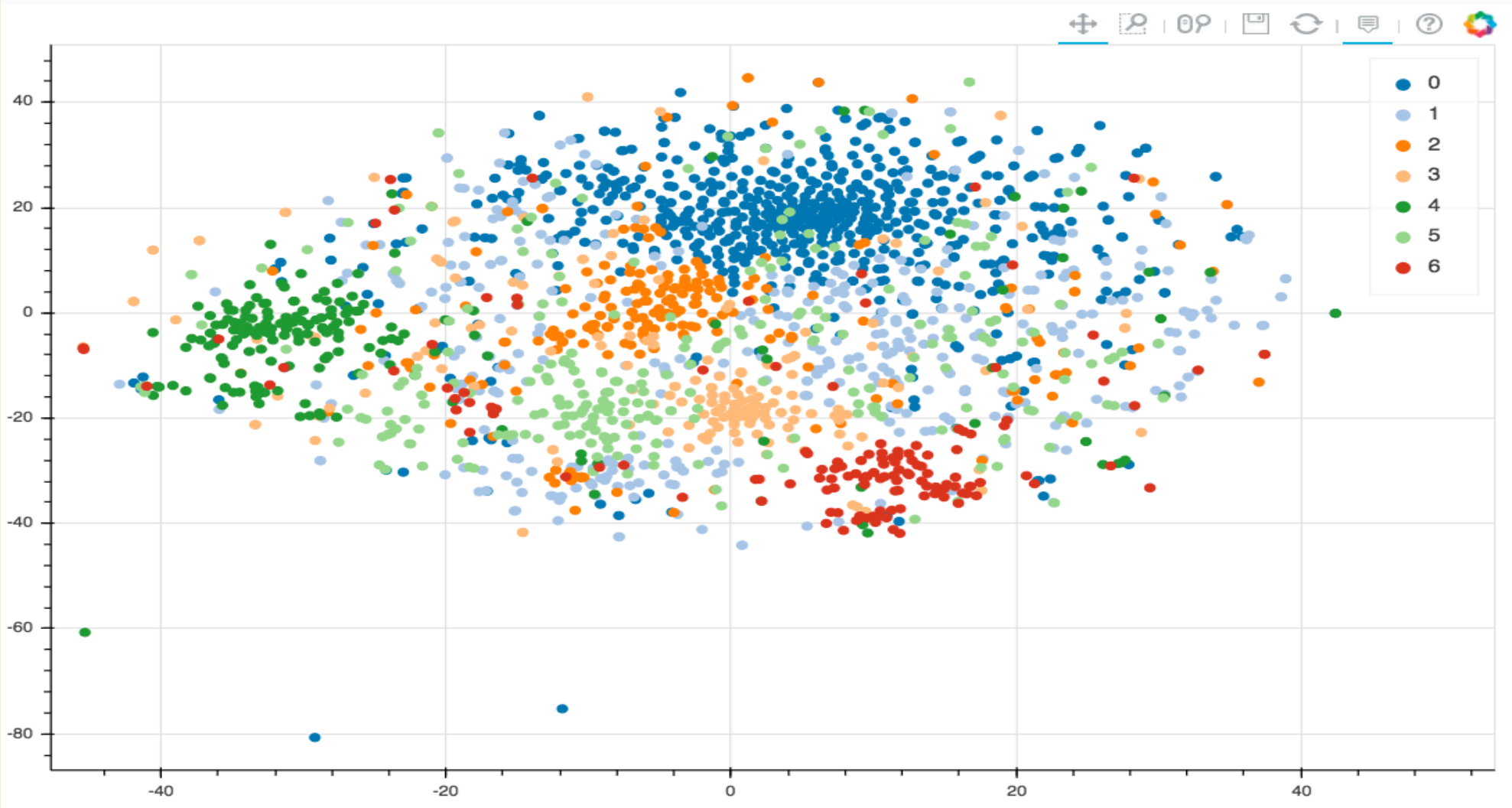
캔버스에 circle 생성

```
from bokeh.models import HoverTool, Legend
from bokeh.palettes import Category20
from bokeh.io import show, output_notebook
from bokeh.plotting import figure, ColumnDataSource
output_notebook()

# 사용할 툴들
p = figure(plot_width=900, plot_height=600,
           toolbar_location='above', x_range=(document_2d[:, 0].min()*1.05, document_2d[:, 1].max()*1.2))

# 각 토픽별 그래프에 추가하도록 source data 생성
circles = []
for k, document_indexes in enumerate(topic_document_indexes):
  document_source = ColumnDataSource(data={
    'x': document_2d[document_indexes, 0],
    'y': document_2d[document_indexes, 1],
    'topic': [k for _ in document_indexes],
    'title': [preprocessed_data[random_index[i]]['title'] for i in document_indexes],
    'body': [preprocessed_data[random_index[i]]['body'][:75] for i in document_indexes],
    'color': [Category20[10][k] for _ in document_indexes],
  })
  circles.append(p.circle('x', 'y', color='color', legend='topic', source=document_source, size=6))

# 몇가지 interaction
p.add_tools(HoverTool(tooltips=[('topic', '@topic'), ("title", "@title"), ('body', '@body')], renderers=circles, mode='mouse'))
p.legend.click_policy = 'hide'
show(p)
```

tooltip interaction 추가

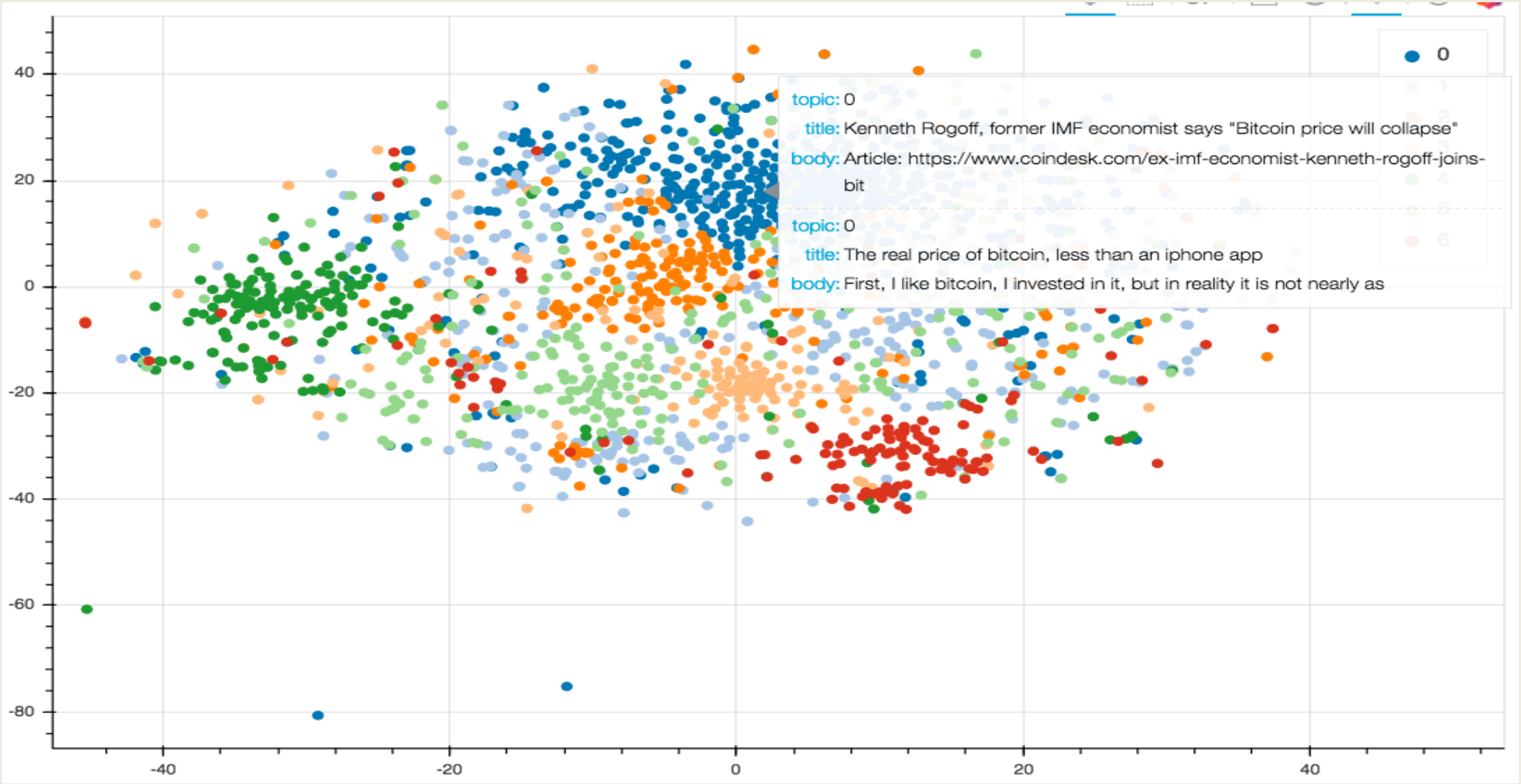
범례를 클릭하면 해당 카테고리의 circle이 가리도록 설정

Results



Results

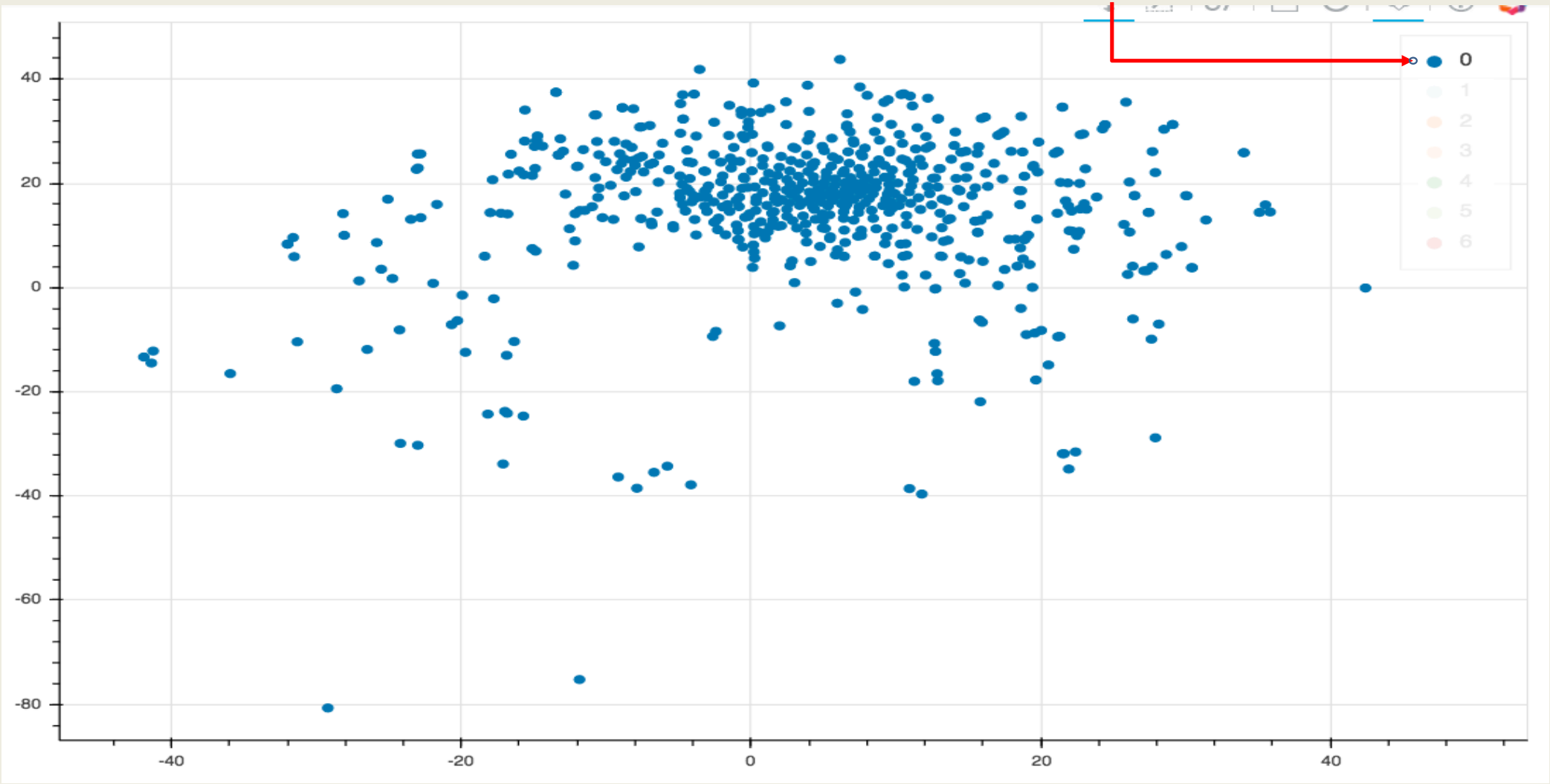
Tooltip interaction
마우스를 갖다대면 해당 데이터의 정보를 tooltip으로



Results

Focussing

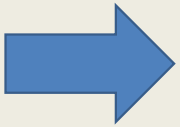
범례의 레이블을 클릭해서 원하는 토픽의 데이터 분포만 확인



Using time series data

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

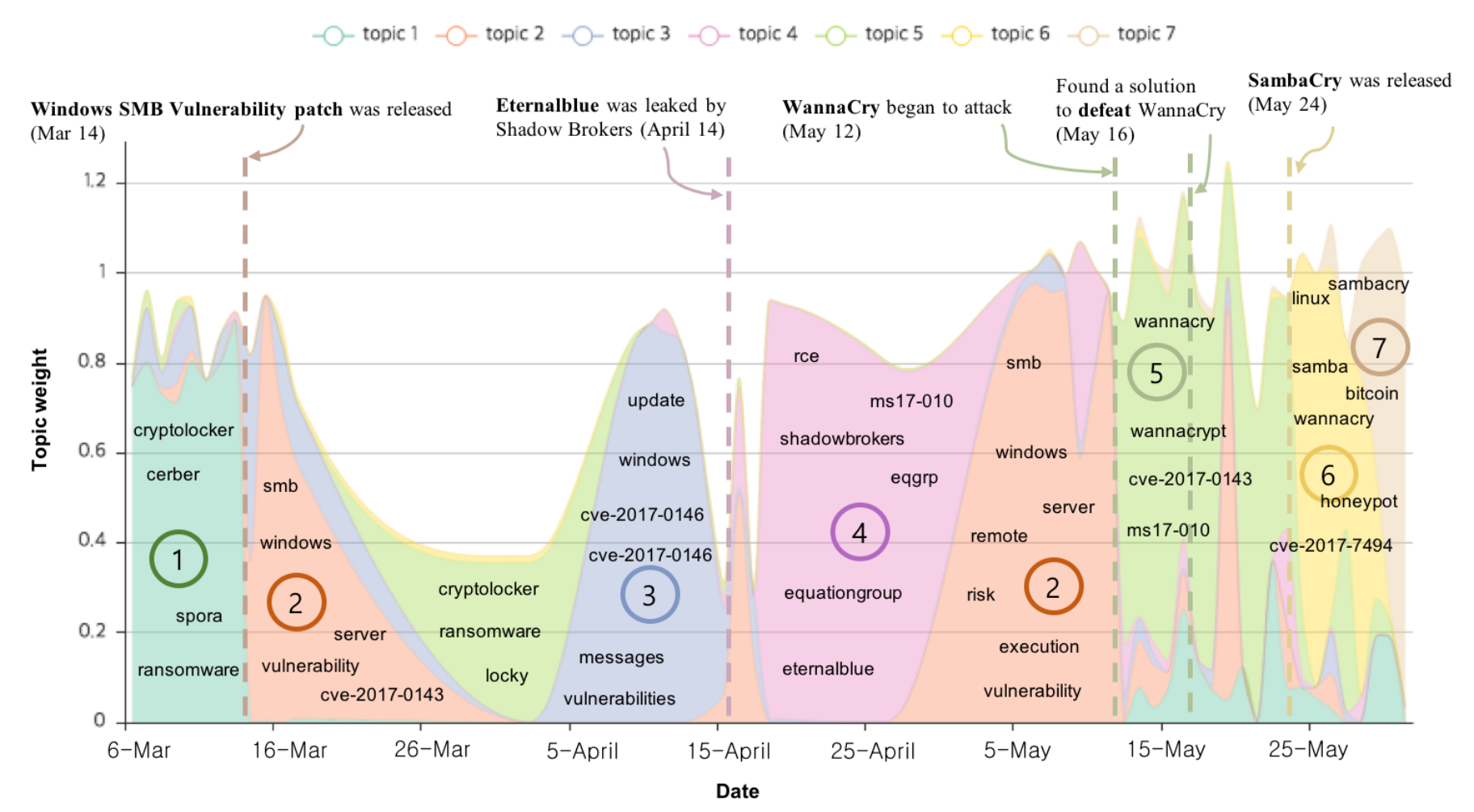
Term-document matrix



	Day1	Day2	Day3
complexity	5	0	1
algorithm	2	3	3
entropy	4	2	7
traffic	1	1	2
network	2	1	4

Term-Time matrix로…

Using time series data




Questions?