
GE17

DATA COLLECTION

Twitter API

Python

Tweepy

PRE-PROCESSING

Parsing

So, what data columns does the Streaming API provide?

```
list(df) # view available columns
```

```
 'user.notifications',
 'user.profile_background_color',
 'user.profile_background_image_url',
 'user.profile_background_image_url_https',
 'user.profile_background_tile',
 'user.profile_banner_url',
 'user.profile_image_url',
 'user.profile_image_url_https',
 'user.profile_link_color',
 'user.profile_sidebar_border_color',
 'user.profile_sidebar_fill_color',
 'user.profile_text_color',
 'user.profile_use_background_image',
 'user.protected',
 'user.screen_name',
 'user.statuses_count',
 'user.time_zone',
 'user.url',
 'user.utc_offset',
 'user.verified']
```

PRE-PROCESSING

Parsing

Cleaning/filtering

Let's keep a set of **sociologically interesting** columns.

```
# get the desired columns by name
cols = list(['created_at']+            # time of tweet
        ['user.name']+          # about the author
        ['user.screen_name']+
        ['user.description']+
        ['user.location']+
        ['user.followers_count']+
        ['user.friends_count']+
        ['in_reply_to_screen_name']+  # addressee
        ['entities.user_mentions.screen_name']+
        ['entities.hashtags.text']+  # tweet contents
        ['entities.media.type']+
        ['entities.media.url']+
        ['text']+
        ['favorite_count']+          # tweet metadata
        #['reply_count']+
        #['retweet_count']+
        #['quote_count']+
        ['lang'])

tweets_df = df[cols]
tweets_df.columns = ['time', 'user', 'user_sn', 'desc', 'loc', 'followers', 'friends',
                     'to_user_sn','mentioning_sns', 'hashtags',
                     'media_type', 'media_url', 'text', 'faves',  'lang']


tweets_df = tweets_df.replace({'\n': ' '}, regex=True) # remove linebreaks in the dataframe
tweets_df = tweets_df.replace({'\t': ' '}, regex=True) # remove tabs in the dataframe
tweets_df = tweets_df.replace({'\r': ' '}, regex=True) # remove carriage return in the dataframe
```

SNA

COLUMNS FOR **SOCIAL NETWORK ANALYSIS**

Let's keep a set of **sociologically interesting** columns.

```
# get the desired columns by name
cols = list(['created_at']+              # time of tweet
        ['user.name']+          # about the author
        ['user.screen_name']+
        ['user.description']+
        ['user.location']+
        ['user.followers_count']+
        ['user.friends_count']+
        ['in_reply_to_screen_name']+  # addressee
        ['entities.user_mentions.screen_name']+
        ['entities.hashtags.text']+  # tweet contents
        ['entities.media.type']+
        ['entities.media.url']+
        ['text']+
        ['favorite_count']+          # tweet metadata
        #['reply_count']+
        #['retweet_count']+
        #['quote_count']+
        ['lang'])
```

```
# Create weighted graph from M
G = nx.Graph()
for u,v,data in M.edges(data=True):
    w = data['weight'] if 'weight' in data else 1.0
    if G.has_edge(u,v):
        G[u][v]['weight'] += w
    else:
        G.add_edge(u, v, weight=w)
```

We now have a graph object:

```
print(nx.info(G))
```

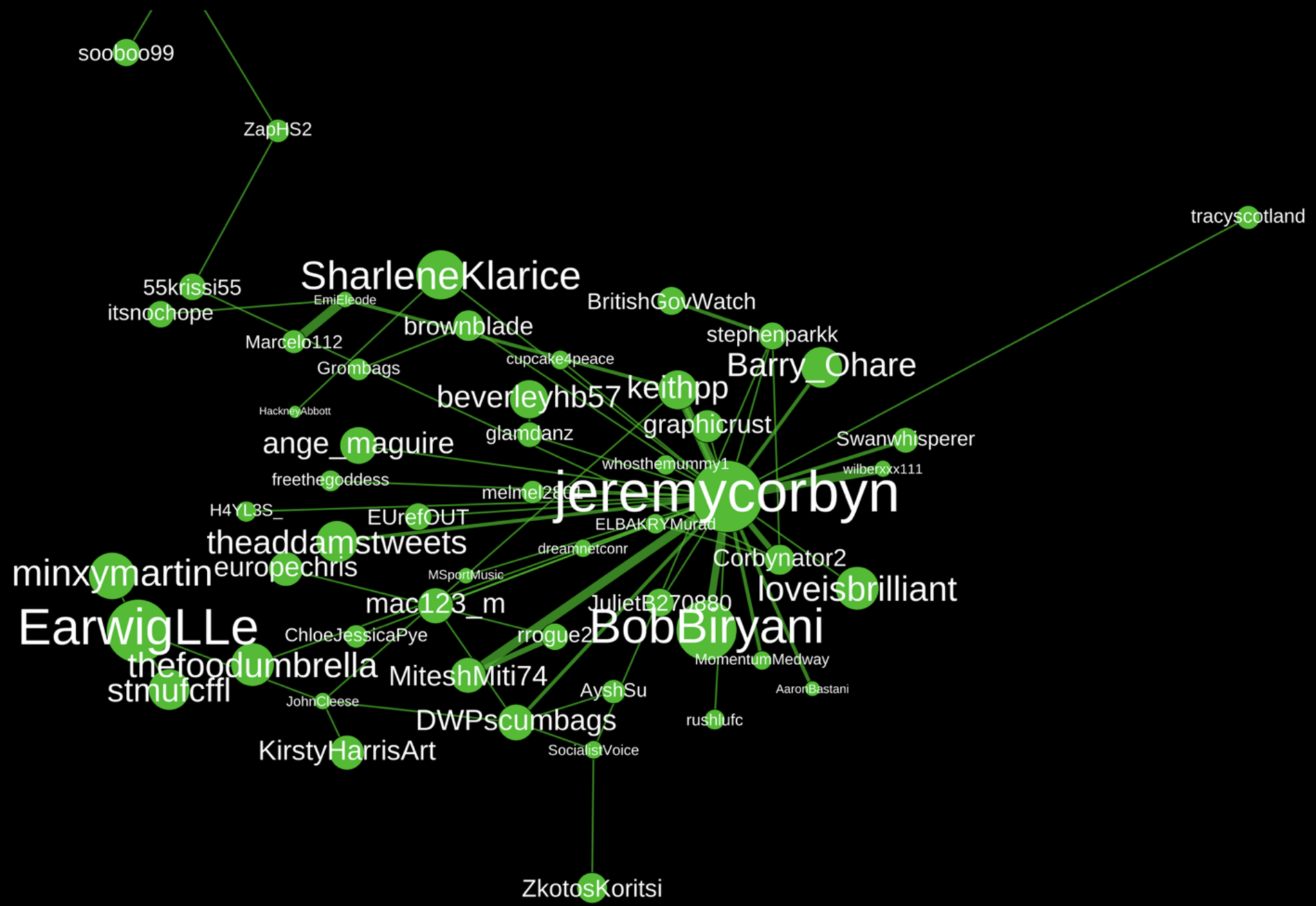
```
Name:
Type: Graph
Number of nodes: 99262
Number of edges: 149684
Average degree:   3.0159
```

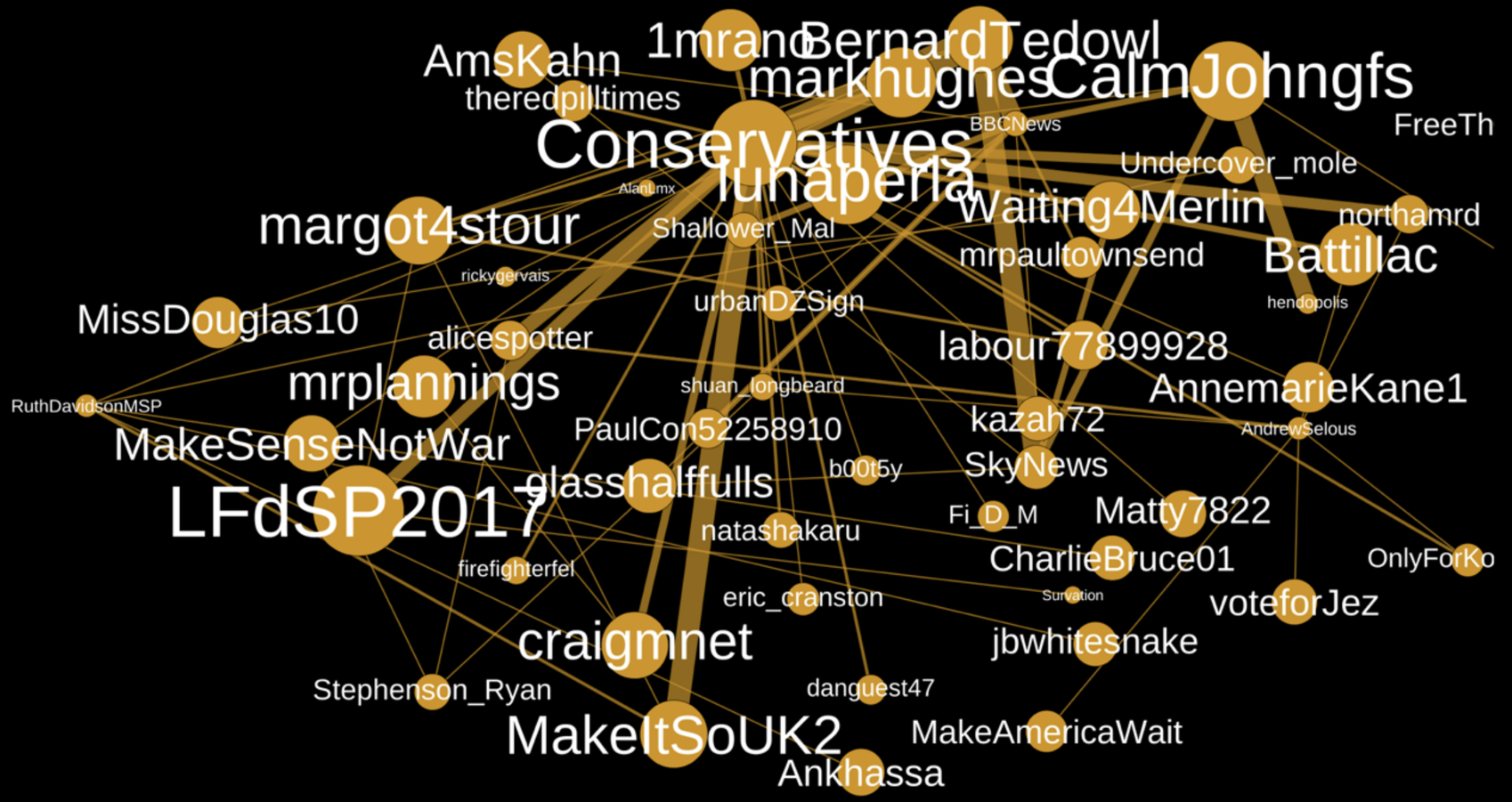
```
# How many subgraphs does G consist of?
nx.number_connected_components(G)
```

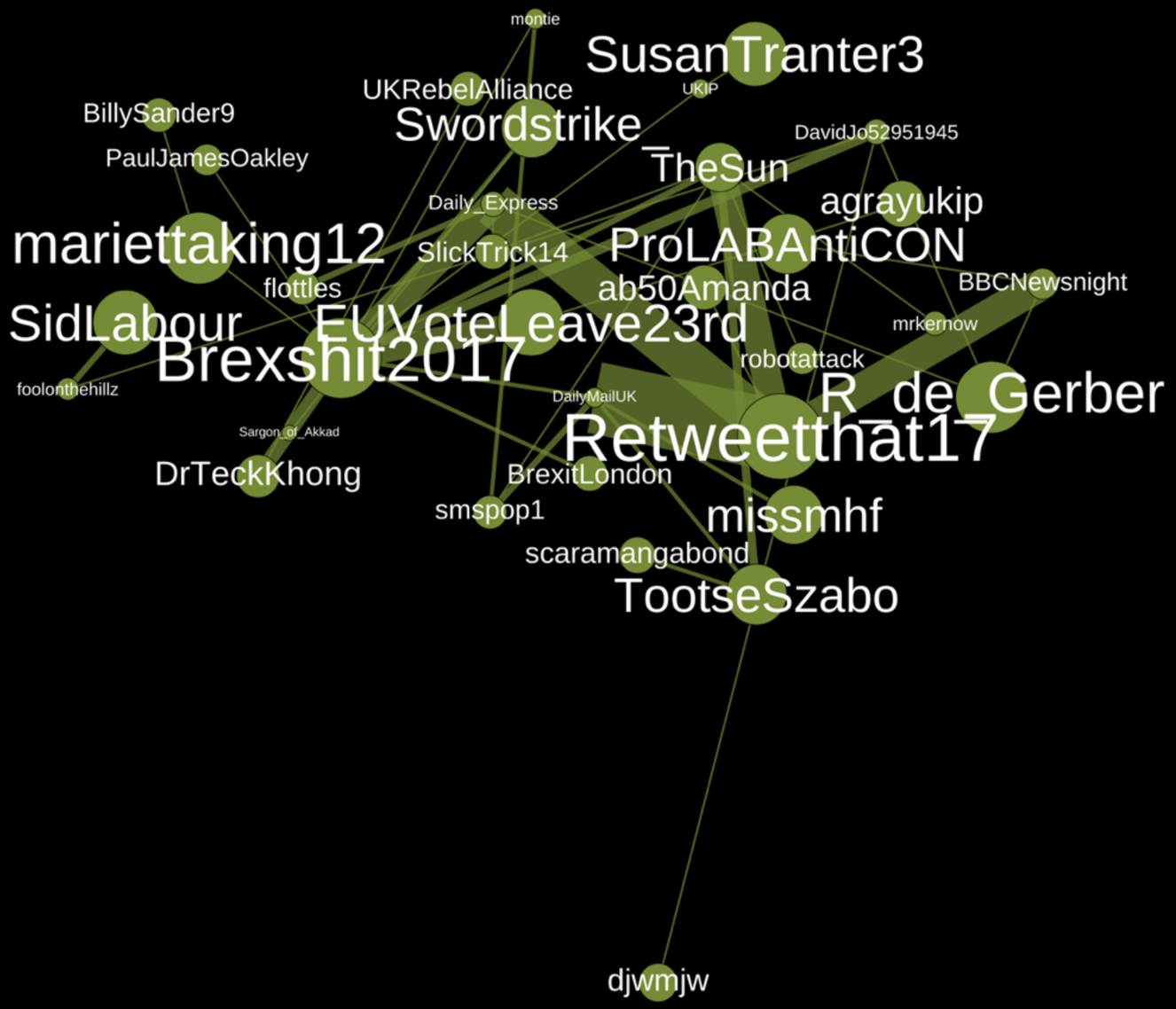
```
2165
```

```
# See sizes of the top 10 connected components
for g in sorted(nx.connected_component_subgraphs(G), key=len, reverse=True)[:10]:
    print(len(g.nodes()))
```

```
94097
39
27
19
19
17
17
14
13
11
```


sooboo99
ZapHS2
tracyscotland
55krissi55
SharleneKlarice
EmiEleode
BritishGovWatch
itsnochope
brownblade
stephenparkk
Marcelo112
cupcake4peace
Barry_Ohare
Grombags
keithpp
beverleyhb57
HackneyAbbott
graphicrust
glamdanz
Swanwhisperer
ange_maguire
whosthemummy1
wilberxxx111
freethegoddess
jeremycorbyn
melmel2801
H4YL3S_
EURefOUT
ELBAKRYMurad
theaddamstweets
dreamnetconr
Corbynator2
minxymartin
europechris
MSportMusic
loveisbrilliant
mac123_m
JulietB270880
EarwigLLe
ChloeJessicaPye
rrogue2
BobBiryani
thefoodumbrella
MomentumMedway
MiteshMiti74
AyshSu
AaronBastani
stmufcffl
JohnCleese
DWPscumbags
rushlufc
KirstyHarrisArt
SocialistVoice
ZkotosKoritsi


1mranon
BernardTedowl
AmsKahn
markhughes
CalmJohngfs
theredpilltimes
BBCNews
FreeTh
Conservatives
Undercover_mole
lunaperla
AlanLmx
Waiting4Merlin
northamrd
margot4stour
Shallower_Mal
mrpaultownsend
Battillac
rickygervais
hendopolis
urbanDZSign
MissDouglas10
alicespotter
labour77899928
mrplannings
shuan_longbeard
AnnemarieKane1
RuthDavidsonMSP
kazah72
AndrewSelous
PaulCon52258910
MakeSenseNotWar
SkyNews
b00t5y
glasshalffulls
LFdSP2017
Fi_D_M
Matty7822
natashakaru
CharlieBruce01
OnlyForKo
firefighterfel
Survation
eric_cranston
voteforJez
craigmnet
jbwhitesnake
Stephenson_Ryan
danguest47
MakeItSoUK2
MakeAmericaWait
Ankhassa


montie
SusanTranter3
UKRebelAlliance
UKIP
BillySander9
Swordstrike_
DavidJo52951945
PaulJamesOakley
TheSun
Daily_Express
agrayukip
mariettaking12
SlickTrick14
ProLABAntiCON
flottles
ab50Amanda
BBCNewsnight
SidLabour
EUVoteLeave23rd
mrkernow
Brexshit2017
robotattack
foolonthehillz
R_de_Gerber
DailyMailUK
Retweetthat17
Sargon_of_Akkad
DrTeckKhong
BrexitLondon
smspop1
missmhf
scaramangabond
TootseSzabo
djwmjw

COLUMN FOR **ANT**

Let's keep a set of **sociologically interesting** columns.

```
# get the desired columns by name
cols = list(['created_at']+              # time of tweet
        ['user.name']+           # about the author
        ['user.screen_name']+
        ['user.description']+
        ['user.location']+
        ['user.followers_count']+
        ['user.friends_count']+
        ['in_reply_to_screen_name']+  # addressee
        ['entities.user_mentions.screen_name']+
        ['entities.hashtags.text']+  # tweet contents
        ['entities.media.type']+
        ['entities.media.url']+
        ['text']+
        ['favorite_count']+          # tweet metadata
        #['reply_count']+
        #['retweet_count']+
        #['quote_count']+
        ['lang'])
```

```
1  RT @CatSmithMP: Look out for this ad van in Fleetwood, Knott End and Lancaster today! #VoteLabour https://t.co/lzegSYZ9x8
2  RT @davidschneider: A personal appeal to the young. From a man who's very much on their wavelength. #GeneralElection17
   https://t.co/1nP5R3I…
3  RT @JeremyCorbyn4PM: The money shot. #TimeForCorbyn #VoteLabour https://t.co/BcgikdbVy5
4  RT @rosa_francesca: remember for tomorrow #GeneralElection17 https://t.co/9UDU5HSdOp
5  I mean I do care. If you have a conscious and like having healthcare available. Vote #labour don't be an idiot.
6  The night before an election do May and Corbyn hear the music that plays the night before the boardroom on the apprentice?
   #GE2017
7  RT @PierceRooney: BBC didn’t want you to see this #Corbyn video. Wonder why #GE17 https://t.co/BPGpGsBKyI
8  @TheSun Did my last bit of campaigning in York Central this evening with @EdYoung4York and @ConservativesYO. Best of luck!
   #GE2017
9  RT @alexnunns: Extraordinary scenes in Islington where #Corbyn has arrived. People chanting "He's coming home, Corbyn's coming
   hom…
10 RT @livlewis63: If you're not voting tomorrow, please keep it to yourself cos actively not caring about your future is
   embarrassing #Genera…
11 RT @Dorsetghost: Vote #Labour tomorrow for a fairer society, No more Austerity, If you're young don't forget to vote
   https://t.co/5tN92iF0V7
12 RT @Harryslaststand: I can't begin to express the pride I have in #JeremyCorbyn &amp;amp; the whole Labour election team for the
   mountains they…
13 Great article about voting choice and whether you are actually making one or being coerced into one of 2 only…
   https://t.co/Yg6LKwvmiZ
14 RT @JamillaTweets: this better not be how we get woken up on Friday morning so go out and vote!!!  #GE2017
   https://t.co/h33TPgvDSD
15 Right-wing parties often make electoral gains after violent attacks, but the #GE2017 isn't as clear cut.…
   https://t.co/1QTqq9eFDL
16 @ShaziaAwan See https://t.co/DdKCOlnHXI for impartial #GeneralElection17 media data analysis. Full report @
   https://t.co/rrcZWGD4bq RT!
17 RT @45albannach: #VoteSNP #GE17 NHS chiefs told to secretly draw up more 'unthinkable cuts' - including to A&amp;amp;E
   https://t.co/5NaUCiHf9a
18 RT @PeoplesMomentum: Powerful Stuff. ⬇️ #VoteLabour #GE2017 https://t.co/6KHRfj6Wup
19 Vote LibDem RT @AlanmakExposed: #Havant #Tory expenses scandal, worth a read. https://t.co/DrkEKAeiNh
20 i now# im annoying u all but it is important that everyone who sees this to vote corbyn tomorrow lets change this country
   #GeneralElection17
```

ThingLinker

Python script by Simon Lindgren // @simonlindgren // simonlindgren.com.

Actor-Network theory (ANT) is about how human and non-human actants are connected in relational systems. It sees entities (humans, texts, machines, activitiees, ideas) as linked to each other in heterogeneous networks. Actors appear in any shape or material. The important thing is not if they have human agency, but whether they have the capacity to cause difference in the course of action of other entitites or not.

One way of analysing processes like these is to look at mechanisms through which an actor is connected to to other actors, and how those other actors in turn are linked to each other. Such analyses can be starting point of making closer assessments of things such as obligatory passage points, interessement, enrolment, and mobilisation). In short, networks are continually made and re-made, by actors who draw links and associations.

In [ ]:
```
# Required Python libraries
import glob, re
import pandas as pd
import spacy
nlp = spacy.load("en") # Set up spaCy with the the default model for English
```

In [6]:
```
dataset = []
data = open('tweets.txt', 'r').readlines()
for l in data:
    dataset.append(l)
```

We then clean the data from things that we do no want. The code below removes urls, any non-alphanumeric characters, double spaces, double line-breaks, and empty lines. Note however, that from the perspective of ANT, things such as urls or emojis can definitely be interesting as actors, so the code below must be customised for the research task at hand.

In [7]:
```
clean = []

for line in dataset:
    line = re.sub(r'(http:\/\/www\.|https:\/\/www\.|http:\/\/|ht
    line = re.sub('[^0-9a-zA-Z]+', ' ', line)
    line = re.sub('  ', ' ', line)
    line = re.sub(r'(\n\n)','\n', line)
    if not len(line.strip()) == 0 :
        clean.append(line)

dataset = clean
```

In [8]:
```
# Inspect the
dataset
```

Out[8]:
```
['RT CatSmithMP Look out for this ad van in Fleetwood Knott E
nd and Lancaster today VoteLabour ',
 'RT davidschneider A personal appeal to the young From a man
who s very much on their wavelength GeneralElection17 ',
 'RT JeremyCorbyn4PM The money shot TimeForCorbyn VoteLabour
',
```

Extracting Things

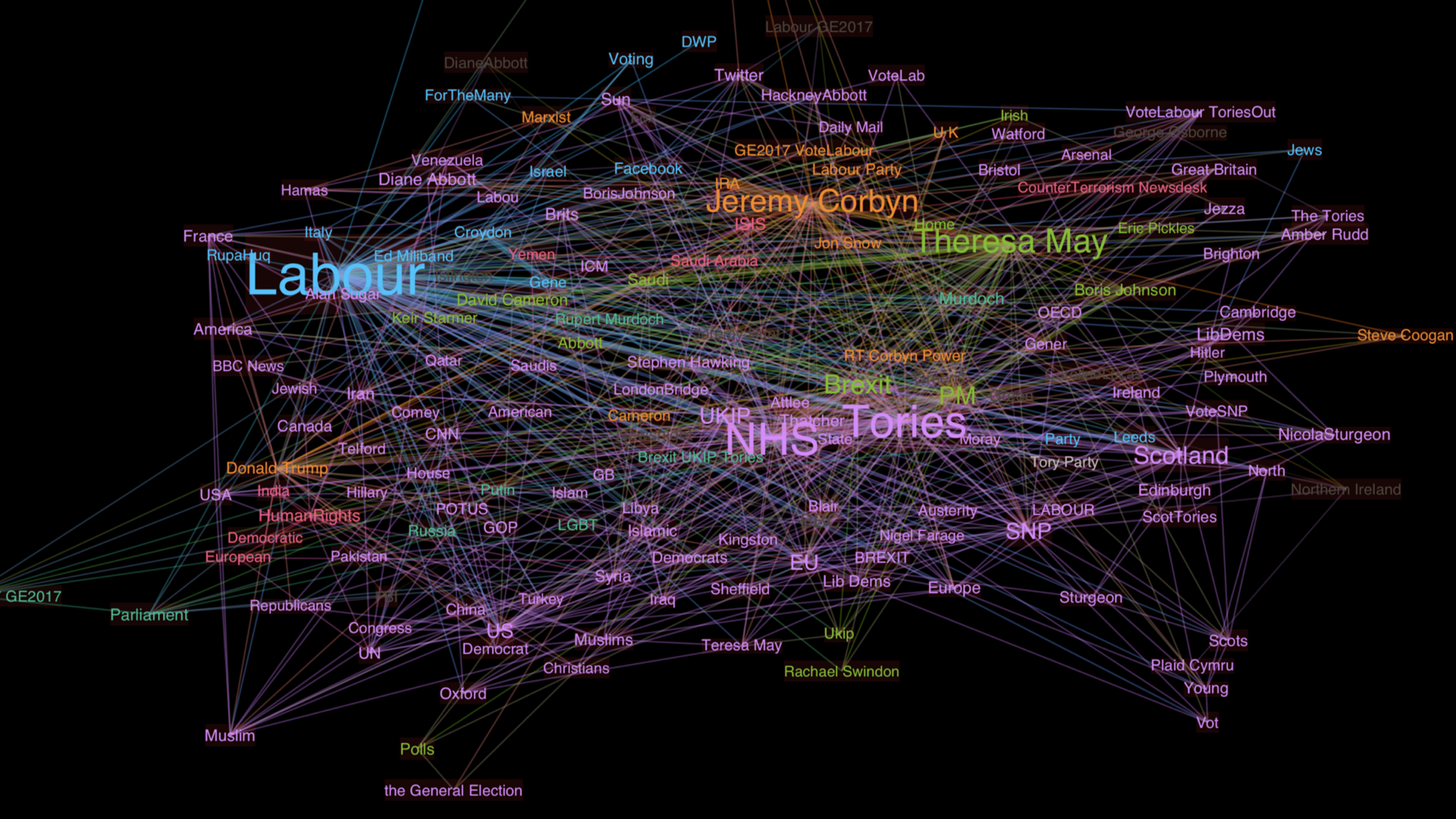
As our next step, we use the `spaCy` Python library to extract what language technologists call 'Named Entities'. These entities – 'Things' – are considered here as potential actors, in the sense of ANT. With `spaCy`, we will get the following tags:

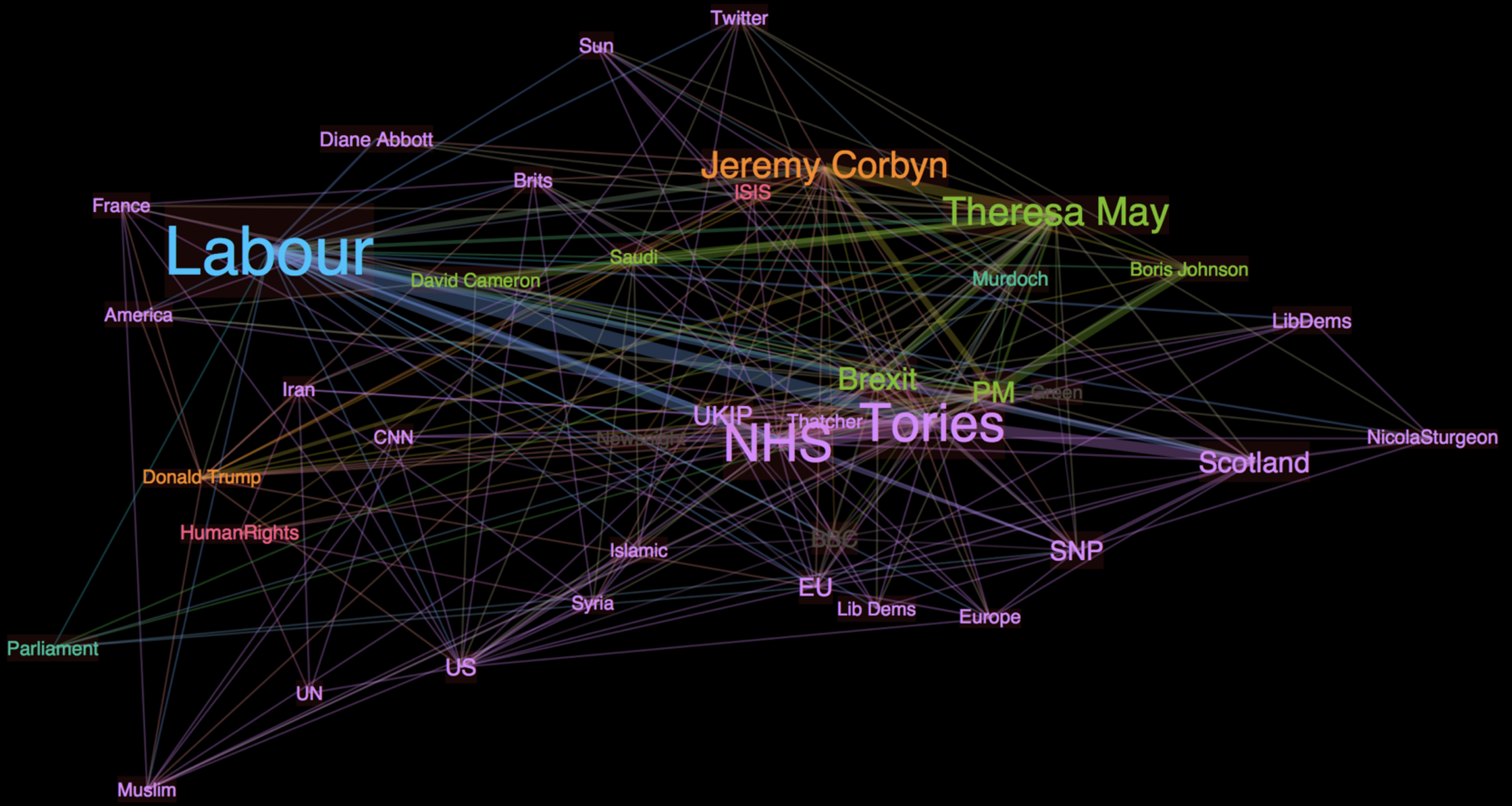
- PERSON People, including fictional.
- NORP Nationalities or religious or political groups.
- FACILITY Buildings, airports, highways, bridges, etc.
- ORG Companies, agencies, institutions, etc.
- GPE Countries, cities, states.
- LOC Non-GPE locations, mountain ranges, bodies of water.
- PRODUCT Objects, vehicles, foods, etc. (Not services.)
- EVENT Named hurricanes, battles, wars, sports events, etc.
- WORK_OF_ART Titles of books, songs, etc.
- LANGUAGE Any named language.

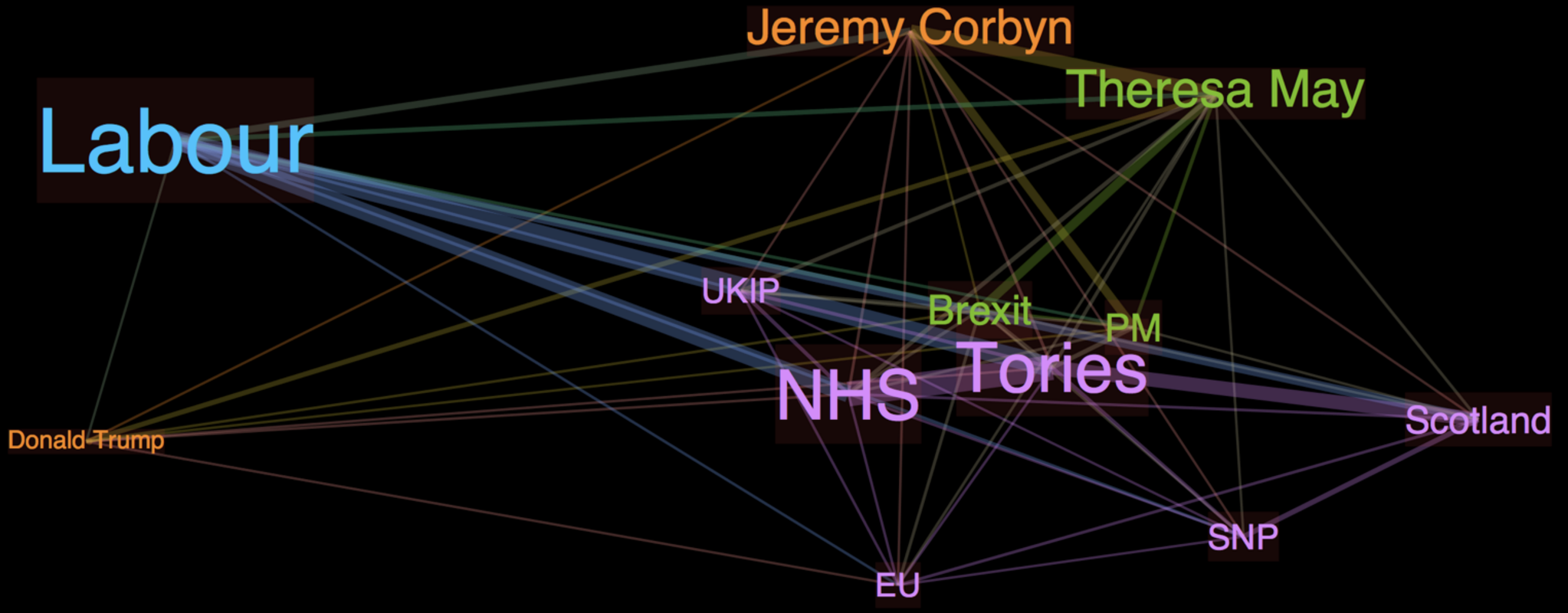
It also extracts the following values:

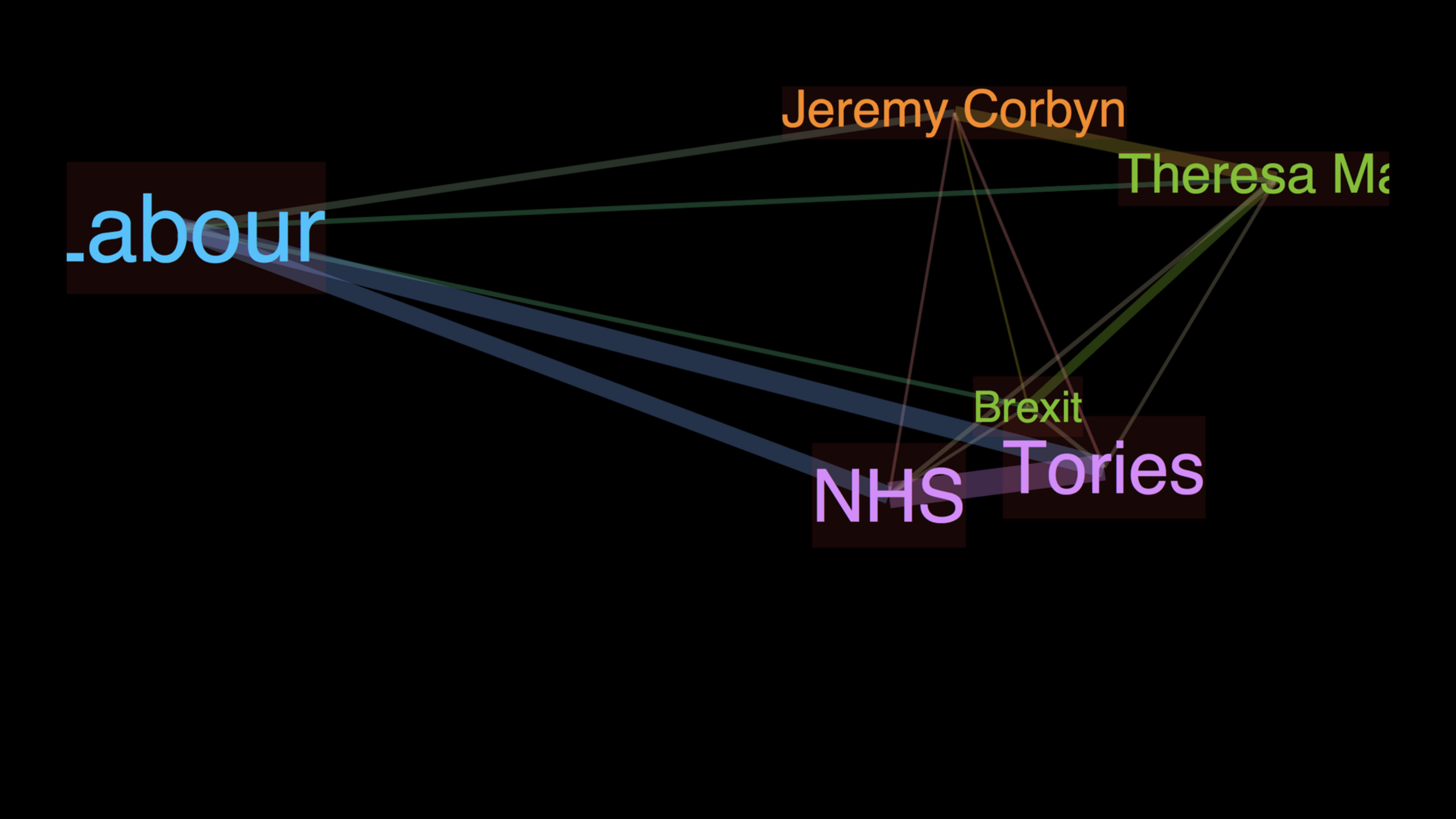
- DATE Absolute or relative dates or periods.
- TIME Times smaller than a day.
- PERCENT Percentage, including "%".
- MONEY Monetary values, including unit.
- QUANTITY Measurements, as of weight or distance.
- ORDINAL "first", "second", etc.
- CARDINAL Numerals that do not fall under another type.

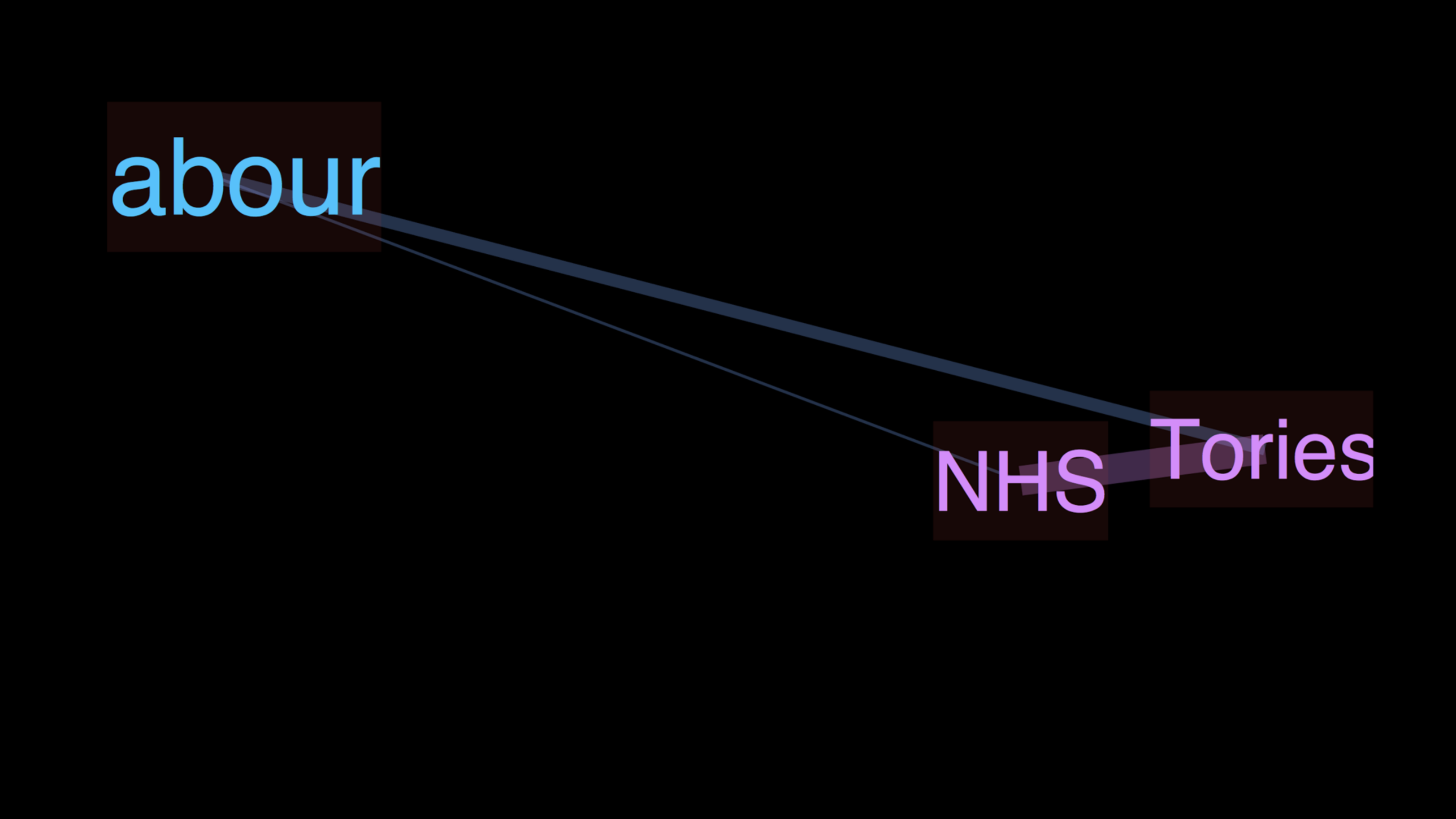
```
198; DATE; tomorrow
198; PERSON; Llanelli
199; CARDINAL; 92
200; CARDINAL; 92
201; ORG; NHS
201; ORG; VoteLabour
201; DATE; tomorrow
201; ORG; NHS
202; NORP; Indian
202; ORG; Lyca
202; CARDINAL; 26
202; ORG; HMR
203; DATE; the 50 days
203; PERSON; Theresa May
205; ORG; Labour
207; DATE; tomorrow
```


Labour GE2017
DWP
Voting
DianeAbbott
Twitter
VoteLab
ForTheMany
Sun
HackneyAbbott
Marxist
Irish
VoteLabour ToriesOut
Daily Mail
U K
Watford
George Osborne
GE2017 VoteLabour
Arsenal
Jews
Venezuela
Israel
Facebook
Labour Party
Bristol
Great Britain
Diane Abbott
IRA
CounterTerrorism Newsdesk
Hamas
Labou
BorisJohnson
Jeremy Corbyn
Jezza
Brits
ISIS
The Tories
France
Italy
Croydon
Home
Eric Pickles
Amber Rudd
Jon Snow
Theresa May
RupaHuq
Ed Miliband
Yemen
Brighton
Labour
ICM
Saudi Arabia
Gene
Saudi
Alan Sugar
David Cameron
Murdoch
Boris Johnson
Keir Starmer
OECD
Cambridge
America
Rupert Murdoch
LibDems
Steve Coogan
Abbott
Gener
Qatar
Saudis
Stephen Hawking
RT Corbyn Power
Hitler
BBC News
Brexit
Plymouth
Jewish
Iran
LondonBridge
PM
Ireland
Attlee
Comey
American
Cameron
UKIP
Thatcher
NHS
Tories
VoteSNP
Canada
CNN
State
Moray
Party
Leeds
NicolaSturgeon
Telford
Brexit UKIP Tories
Tory Party
Scotland
Donald Trump
House
GB
North
USA
India
Hillary
Putin
Islam
Edinburgh
Northern Ireland
HumanRights
POTUS
Libya
Blair
Austerity
LABOUR
ScotTories
Democratic
Russia
GOP
LGBT
Islamic
SNP
Kingston
Nigel Farage
European
Pakistan
Democrats
EU
BREXIT
Syria
Lib Dems
GE2017
Sheffield
Europe
Sturgeon
Republicans
Turkey
Iraq
Parliament
China
Congress
US
Ukip
Muslims
Scots
UN
Democrat
Teresa May
Plaid Cymru
Christians
Rachael Swindon
Young
Oxford
Vot
Muslim
Polls
the General Election


Twitter
Sun
Diane Abbott
Jeremy Corbyn
Brits
ISIS
France
Theresa May
Labour
Saudi
David Cameron
Murdoch
Boris Johnson
America
LibDems
Brexit
Iran
PM
Green
UKIP
Thatcher
Tories
CNN
NHS
NicolaSturgeon
Scotland
Donald Trump
HumanRights
BBC
Islamic
SNP
EU
Syria
Lib Dems
Europe
Parliament
US
UN
Muslim


Jeremy Corbyn
Theresa May
Labour
UKIP
Brexit
PM
NHS
Tories
Scotland
Donald Trump
SNP
EU


Jeremy Corbyn
Theresa Ma
abour
Brexit
NHS
Tories


abour
NHS
Tories