# Mapping the echo-chamber *COVID-19 edition*

Manuel Ivagnes
Riccardo Bianchini
Valerio Coretti

La Sapienza University of Rome — July 23, 2020

# Intro & Related work

*“An echo chamber is a metaphorical description of a situation in which beliefs are amplified or reinforced by communication and repetition inside a closed system and insulates them from rebuttal.”* - wikipedia

**Experiment:**

- **Goal:** Find the polarization of communities inside the *Twitter* social network regarding Coronavirus related topics
- **How:** Network graph ⇒ Communities ⇒ Semantic deviation

**Reference papers:**


- Armineh Nourbakhsh, Xiaomo Liu, Quanzhi Li, Sameema Shah, **Mapping the echo-chamber: detecting and characterizing partisan networks on Twitter**, Research and Development, Thomson Reuters
- Hywel T.P. Williams, James R. McMurray, Tim Kurz, F. Hugo Lambert, **Network analysis reveals open forums and echo chambers in social media discussions of climate change**, Global Environmental Change, Volume 32, 2015, Pages 126-138, ISSN 0959-3780
- Svetlana S. Bodrunova, Ivan S. Blekanov, Mikhail Kukarkin, **Language and Sentiment Structure of Twitter Discussions on the Charlie Hebdo Case**, from *HCI International 2018 – Posters’ Extended Abstracts: 20th International Conferences*, HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part I

# Dataset & preprocessing

**Seed:** The "**Coronavirus Tweet Ids**" dataset from Harvard University [V1][1]

- 51,798,932 tweets ids related to Coronavirus or COVID-19;
- From March 3, 2020 to March 19, 2020;
- Keywords: #Coronavirus, #Coronaoutbreak, #COVID19;

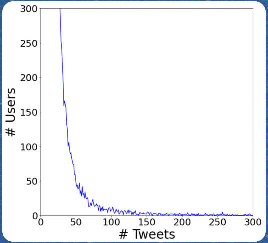
**Final:** 2,480,875 complete tweets from 1,169,150 unique accounts

1. Sample 1 tweet ID each 10
2. Hydrate ids (Twarc)
3. Lossy compression
4. Only English
5. List of all the users ⇒ *accounts.jsonl*

HARVARD
Dataverse

[1] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/LW0BTB

# Statistics




## 90% of accounts shared less than 4 tweets

- 1 tweet ⇒ 795383 users
- 2 tweets ⇒ 186736 users
- 3 tweets ⇒ 68835 users

## Few accounts shared many tweets

- 816 tweets ⇒ 1 users
- 1067 tweets ⇒ 1 users
- 1917 tweets ⇒ 1 users

## Most popular domains do not represent relevant connections

```
#1 => twitter.com: 239036
#2 => bit.ly: 23535
#3 => youtu.be: 7186
#4 => paper.li: 5696
#5 => ow.ly: 4154
#6 => www.instagram.com: 4079
#7 => www.pscp.tv: 3985
#8 => buff.ly: 3101
#9 => trib.al: 2919
#10 => www.youtube.com: 2150
```

```
#1 => https://twitter.com/i/events/1219057585707315201: 2720
#2 => https://twitter.com/maxbrooksauthor/status/1239624352305303552: 2537
#3 => https://twitter.com/BrookeGMcDonald/status/1238986272137502720: 2100
#4 => https://twitter.com/Reuters/status/1239637550828064769: 1761
#5 => http://bit.ly/337yabc: 1702
#6 => https://trib.al/vVSjvun: 1438
#7 => https://twitter.com/messages/compose?recipient_id=835740314006511618&
#8 => https://twitter.com/redfishstream/status/1238436668102893568: 1210
#9 => https://twitter.com/LizSpecht/status/1236095180459003909: 944
#10 => https://twitter.com/adamclarkitv/status/1236289649737371648: 934
```

# Network graph

Step-by-step construction (node:*user*, edge:*connection*):

1. *sampled_accounts.jsonl* ⇒ All the users sampled (30,000)
   *sampled_tweets.jsonl* ⇒ All the corresponding tweets
2. *inverted_domains.jsonl* ⇒ Pseudo inverted index $<$ *domain* : *users* $>$
   For 10 most popular domains all the url of the page
3. *inverted_hashtags.jsonl* ⇒ Pseudo inverted index $<$ *hashtag* : *users* $>$
   Tags similar to dataset keywords removed
4. *retweet_mentions.jsonl* ⇒ For each user all the users mentioned and retweet connection $<$ *user* : *connections* $>$

```
Initial                          # nodes: 0,     # edges: 0
After domains                    # nodes: 2995,  # edges: 18252,   avg. clustering coeff: 0.731353829365398
After hashtags                   # nodes: 10298, # edges: 441348,  avg. clustering coeff: 0.779130029889205
After retweets/mentions          # nodes: 10938, # edges: 442391,  avg. clustering coeff: 0.7294592359119989
```

5. Extract *Giant component*

```
Giant component                  # nodes: 10162, # edges: 441707,  avg. clustering coeff: 0.7613322629146598
```

# Communities detection

Louvain method tries to identify communities by maximizing modularity [2]:

$$Q(G,S) = \underbrace{\frac{1}{2m}}_{-1<Q<1} \sum_{s\in S}\sum_{i\in S}\sum_{j\in S}\left(A_{ij} - \frac{k_i k_j}{2m}\right)$$

**1st Attempt: Louvain method - From scratch**

- The algorithm extract 4879 communities
- May seem good result but NOT optimal partitioning!

**2nd Attempt: Louvain method - Python community API**

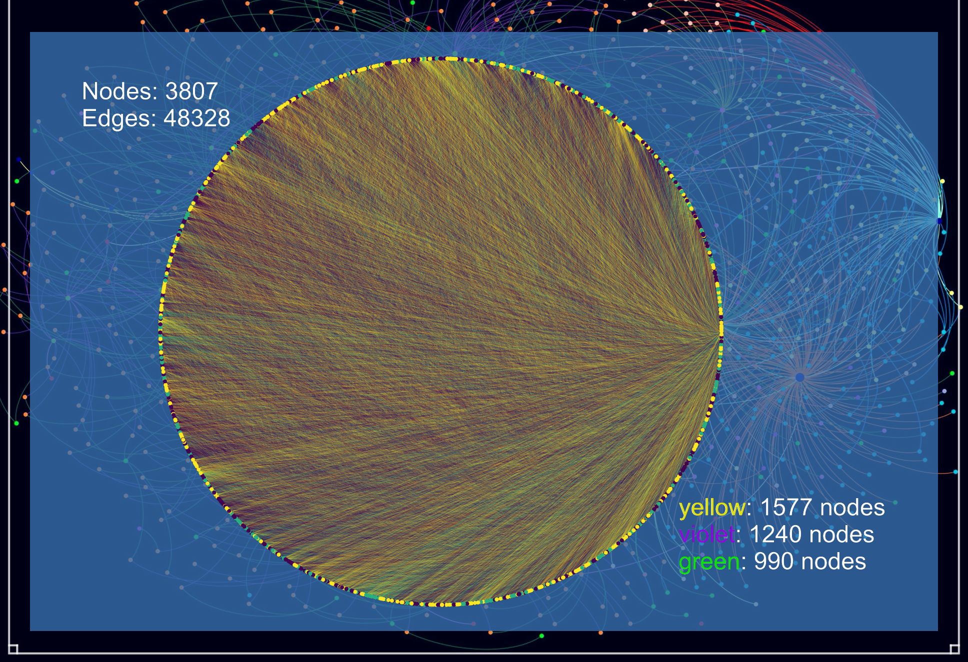
- The algorithm extract 39 communities
- Extracted the 3 largest community to draw the circular graph

**Reference paper:**

- Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, **Fast Unfolding of communities in large networks**

[2] CS246: MMD, Jure Leskovec, Stanford University - Community Detection in Graphs: http://cs246.stanford.edu

# Communities circular graph



# Word2vec preprocessing

**Why:**

- Tweets corpus may contains URLs, hashtags and so on ⇒ specific tweet text cleaning required to obtain normal text

| | |
|---|---|
| 0 | "BREAKING: The University of Liverpool has clo... |
| 1 | I am forever grateful to live in this city. It... |
| 2 | #coronavirus https://t.co/9o570wMHF7" |

**Steps (for each community):**

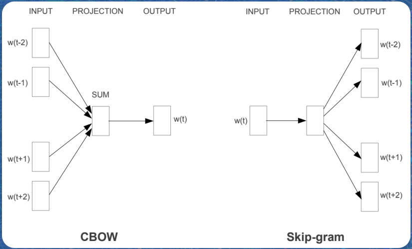
1. Remove all duplicates and null text values
2. Remove emoticons, URLs and hashtags
3. Classical text pre-processing, such as removing stop words and alphanumerical words, applying lower-casing, removing punctuation and extra white spaces
4. Remove resulting empty rows;

| | |
|---|---|
| 0 | breaking the university liverpool closed toda... |
| 1 | forever grateful live city right thing scouse... |

# Word embedding & Word2vec

**Word embedding** ⇒ set of language modeling and feature learning techniques in NLP where words are mapped to vectors of real numbers

**Word2vec** ⇒ group of related models that are used to produce word embeddings. Implemented as two-layer neural networks and trained to reconstruct linguistic contexts of words




**Reference paper:**


- Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, *Efficient Estimation of Word Representations in Vector Spaces*

# Semantic Deviation

**Input**: *word embeddings*
**Output**: *deviation* of every term

**Global dataset**: set of pre-trained vectors created on a corpus of 198 million tweets[3]

**Semantic deviation**:

- Deviation of a term $t$ from community $c$:

$$dev(t, c) = cosine\_distance(v_t^c, v_t^g)$$

- Community with the largest deviation for that term:

$$max\_dev(t) = argmax_c \; dev(t, c)$$

3 https://zenodo.org/record/581402

# Results

Terms that show marked deviation in a certain community, but not in others.

- 10 most deviated terms in each community:

| Com 0 | Com 1 | Com 2 |
|---|---|---|
| ('including', 1.1544819921255112) | ('australia', 1.1779276877641678) | ('lombardy', 1.1795322597026825) |
| ('billion', 1.1469375491142273) | ('announces', 1.1319914907217026) | ('deal', 1.176911398768425) |
| ('congress', 1.1316450238227844) | ('ceo', 1.12015251070261) | ('iran', 1.1741150617599487) |
| ('others', 1.1289057284593582) | ('private', 1.1160971522331238) | ('aggressive', 1.1442659497261047) |
| ('call', 1.1269844621419907) | ('india', 1.110935539007187) | ('word', 1.1383947730064392) |
| ('wouldnt', 1.1232784315943718) | ('option', 1.108831726014614) | ('left', 1.1374759674072266) |
| ('request', 1.11934744566679) | ('nothing', 1.1030820235610008) | ('senate', 1.1324164420366287) |
| ('april', 1.1176810264587402) | ('ill', 1.1021325066685677) | ('flights', 1.132319375872612) |
| ('claim', 1.1157629638910294) | ('goes', 1.102097101509571) | ('false', 1.1317063122987747) |
| ('war', 1.1152003332972527) | ('distancing', 1.1016505435109138) | ('hit', 1.1295667588710785) |
| Yellow | Green | Violet |

- Defined as clusters of 3 neighboring terms (namely those with the highest similarity) that all have the highest distance from the global vector.

| Community | Yellow | Green | Violet |
|---|---|---|---|
| **Deviated Terms** | billion, telling, respond<br>italians, decision, fighting<br>wuhan, germany, feb | australia, positive, disease<br>india, flu, things<br>smart, negative, epidemic | flights, word, small<br>country, iran, nation<br>lombardy, director, later |

# Conclusion

In conclusion, our experiment confirmed the followings:

- We can identify latent networks or communities making echo-chambers inside the Twitter social network without any need of supervision;
- For a given echo-chamber, we can automatically identify topics or phrases that the community is vulnerable to spreading misinformation about, by analyzing the distribution of vector representations of messages;

Possible future work:

- Expand the dataset;
- Increase the number of accounts considered to make the graph;
- Classification of users attitudes and sentiment;

**THANK YOU**