# Power of BTS ARMY for Social Change Envisaged by Twitter Network Analysis

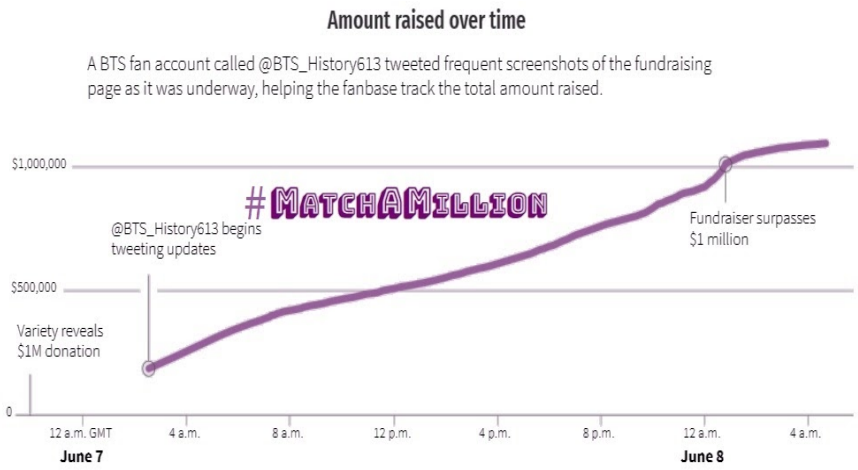
Vrinda, Inhee & Anirudh
https://github.com/TheChirpyWitch/BLMAndBTSArmy

## I. INTRODUCTION

In June 2020, BTS (South Korean hip-hop boy group) donated one million dollars for Black Lives Matter (BLM). The BTS official fanclub, called ARMY, had mobilised via Twitter and decided to match the donation. They used Twitter hashtags #MatchAMillion, #Match1Million and #MatchTheMillion to spread word about the campaign. Within 24 hours between 7th-8th of June, 2020, the fundraising account @OneInAnARMY announced that they had met their goal of 1 million dollars for BLM. `[bts1,bts2]`
BTS Army is known for supporting various causes, so the initial campaign was not surprising. What was surprising was the speed at which this feat had been completed. This 24 hour window became the focal point of our analysis.

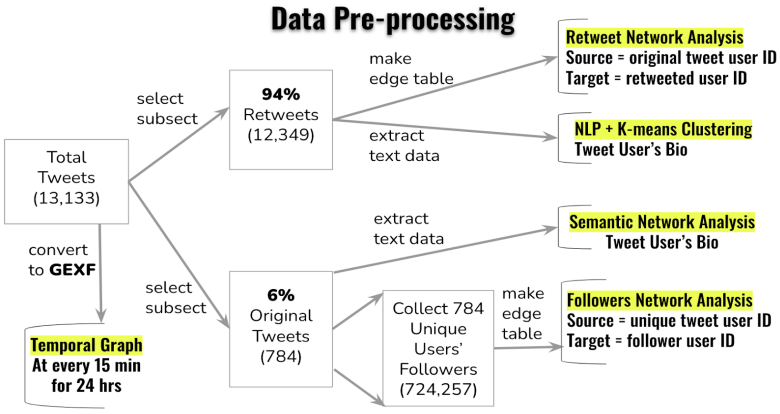
## II. ALGORITHMS & EXPERIMENTS

### II-1) Twitter Data Crawling of #MatchMillion for 24 Hours




All three team members have applied to the Twitter API Developer Accounts, and have been granted the required credentials. We have scrapped the Twitter data for one hashtag #MatchAMillion for 24 hours time windows at every 15 minutes interval between 2020-06-07-2020-06-08 using python script. Total of 13,133 data points collected.

### II-2) Data Pre-processing




For subsequent structural and behavioral data analysis from the network graphs, we perform the various data pre-processing depicted in the figure below.
Firstly, the entire 13,133 tweets data points are converted to Gexf format for temploral graph construction.
Secondly, out of the entire tweets data, 94% is retweets. From these, we make edge tables

consisting of two columns of source nodes and target nodes, and construct a retweet network graph. In addition, we extract the users' bio description text data for NLP analysis.
Thirdly, only about 6 % is original tweets out of the entire dataset. So, from these original tweets, we extract the users' bio description text data for semantic network analysis.
Lastly, from the identified 784 unique twitter users ID (writers of original tweets), we further collect those users' followers information totaling 724,257 data points. From these followers' data, we make an edge table consisting of two columns, one is source ID (original tweet writer's ID), another is target ID (source ID's follower ID) to construct and analyze the follower network.

### II-3) Networks Structural Analysis: Network Graph & Community Detection by Louvain Algorithm

To build a social network we prepared the "Edge" table data: connectivities between users (source and target) via retweet as well as leader-followers serve as directed Edge information. We then import this edge table to the Gephi. For community detection, the default community detection algorithm in Gephi is Louvain Algorithm [louvain] designed to extract the community structure of large networks. It is a heuristic method based on modularity optimization by two iterative phases: 1) modularity optimization by allowing only local changes of communities; 2) aggregation of the found communities to build a new network of communities.

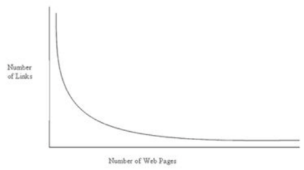
### II-4) Dynamic Network Analysis

We use the Gaphi 0.9.2 for constructing dynamic temporal network graphs. Firstly, we convert the collected 24 hours twitter data in JSNL format to GEXF (Graph Exchange XML Format), which can be imported to the Dynamic/Temporal Network module of the Gephi. We then click the "Overview" button, followed by "Statistics" button to compute structural statistics, especially to compute "modularity" (resolution 3.0 or so to have ~10 major communities with > 1% population). Resulting community attributions are mapped on to the network graph by "modularity class" in different colors, as well as the size of nodes are represented by the computed "out-degrees". We then enable the time variations to slide the time window from 0 to 24 hours at every one hour interval.

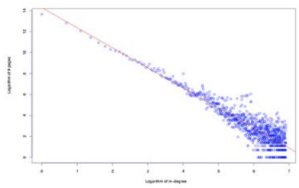
### II-5) Information Cascade Analysis

### II-5a) Power Law Analysis

$$y \propto x^{-k}$$ Where, $K > 1$




Power Law plot



Log-log plot of Power Law

**Scale-free network:** "Scale-free networks' structure and dynamics are independen of the system's size *N*, the number of nodes the system has."

-Wikipedia

Power law is the mathematical representation of "rich get richer" philosophy. Most real world or scale-free networks follow power law. We did power law analysis for both networks.

- Retweet network -> plot of number of retweets vs. number of nodes with that number of retweets
- Followers -> plot of number of followers vs. number of nodes with that number of followers

We found the threshold for each power law graph and compared it with the normal twitter network's power law threshold. We used powerRLaw and ggplot libraries for the same.

### II-5b) Common Neighbors Algorithm

**User Engagement:** Relation between static followers and dynamic real-time retweets.

**General Common Neighbors Algorithm**

- Based on similarity between neighborhoods of two nodes u and v
- Similarity(u, v) = $|\Gamma(u) \cap \Gamma(v)|$
- $\Gamma(u)$ and $\Gamma(v)$ are neighbours of u and v

**Adapted Common Neighbors Algorithm**

Based on similarity between neighborhoods of the same node in two networks

Similarity($u_i$, $v_i$) = $|\Gamma(u_i) \cap \Gamma(v_i)|$

$\Gamma(u_i)$: neighborhood of node i in retweets network

$\Gamma(v_i)$: neighborhood of node i in followers network

$\Gamma(u_i) \cap \Gamma(v_i)$: number of followers who retweeted

User engagement = $\frac{\text{No. of retweets by followers}}{\text{No. of followers}}$

### II-6) Linguistic Analysis Method:

### II-6a) Word Clouds

Simply put, word clouds are a way to visualize word frequencies. We used R's tm package `[tm]` to clean and remove stopwords, punctuations, etc. of user_bio and the tweets corpus and used wordcloud and RColorBrewer to plot the word clouds. `[wcloud]`

### II-6b) Sentiment Analysis

Sentiment analysis tells us if the text is positive or negative. We used the twitter_samples(it has positive and negative tweet samples) from Python's NLTK package `[nltk]` to train on a Naive Bayes model. Then, we ran the model on the cleaned dataset from the previous section.

### II-6c) Word2Vec

Basically, we convert retweet user's bio description to numeric vectors using Word2Vec as an NLP embedding method to prepare for the numeric data, so that we can further analyze them using Unsupervised Machine Learning techniques such as k-means clustering in a reduced dimensional space.

Extracted user's bio text data is firstly lemmatized (e.g. talking→ talk) and tokenized (split a sentence into a word list). We then convert text tokens into numerical vectors using text embeddings APIs, namely TF-hub and its wrapped version of "Embeddings-As-Service". `[embeddings]` TF-hub is a consolidated NLP hub where pre-trained various NLP embedding models are available. For more convenience, the models in TF-hub are further wrapped in the form of python API, so that we can just input our text, then can get the embedded numeric vectors without any training.

### II-6d) Semantic Network

A semantic network, or frame network is a knowledge base that represents semantic relations between concepts in a network. This is often used as a form of knowledge representation. We used this to find themes

within the corpus. We used the cleaned data from II-6a), selected the words that were in the top 80% of the dataset and created a similarity matrix for the words, treating each word as a document. The cosine similarity would become the weights for the network. Then, we ran igraph's cluster_walktrap to find clusters within the network. These clusters gave us the themes for the corpus. `[igraph]`

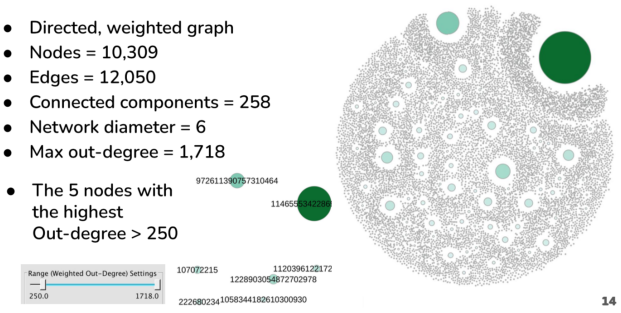
### II-6e) Classification

We gave each theme from the previous section weights and used available BTS ARMY resources to create a list of markers for ARMY accounts. For each user's bio, we checked for the presence of these markers, and summed the weights. According to the final weight, we assigned a simple "fan or not" flag.

## III. RESULTS & DISCUSSION

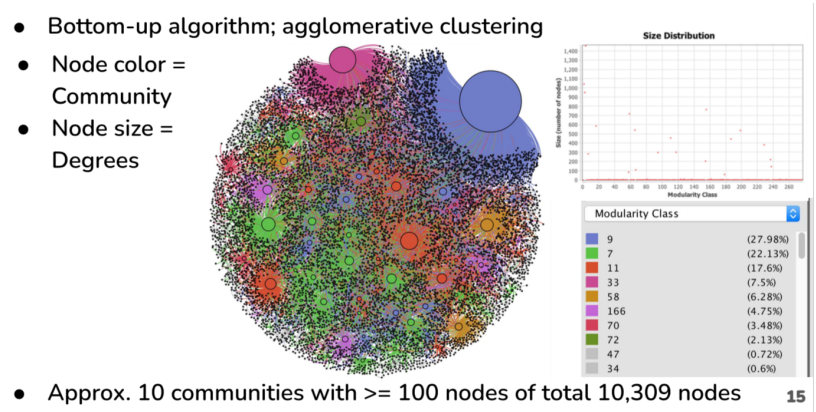
### III-1) Retweet Network Structural Analysis

Construction & Characterization of the Network




Retweet data based network is shown left, consisting of 10K nodes and 12K edges. Total number of connected components is 258. When we filter the out-degree larger than 250, those top 5 nodes (i.e. tweet user ID) are identified. Later in power analysis, we can identify them further as the BTS-fan account or not.

Community Detection & Modularity

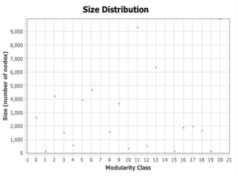



We have experimented with different resolution numbers to adjust the resulting number of distinct communities. We found that using a resolution of 3.0, we can obtain approximately 10 distinct communities which of each consists of at least 100 nodes. The community detection figure shown left represents distinct communities in different colors and its node side represents out-degree.

### III-2) Followers Network Structural Analysis

Due to limitation of memory, instead of importing entire followers edge tables, we randomly select 100 unique users and their followers for edge data. We did such random selection twice to generalize some patterns observed in the followers network.

- Randomly select 100 users (original tweet writers) and their followers
- Nodes = 55,638 | 70,537
- Edges = 65,697 | 80,073
- Connected components = 4 | 5
- Network diameter = 3 | 3
- Node color = Community
- Node size = Degrees

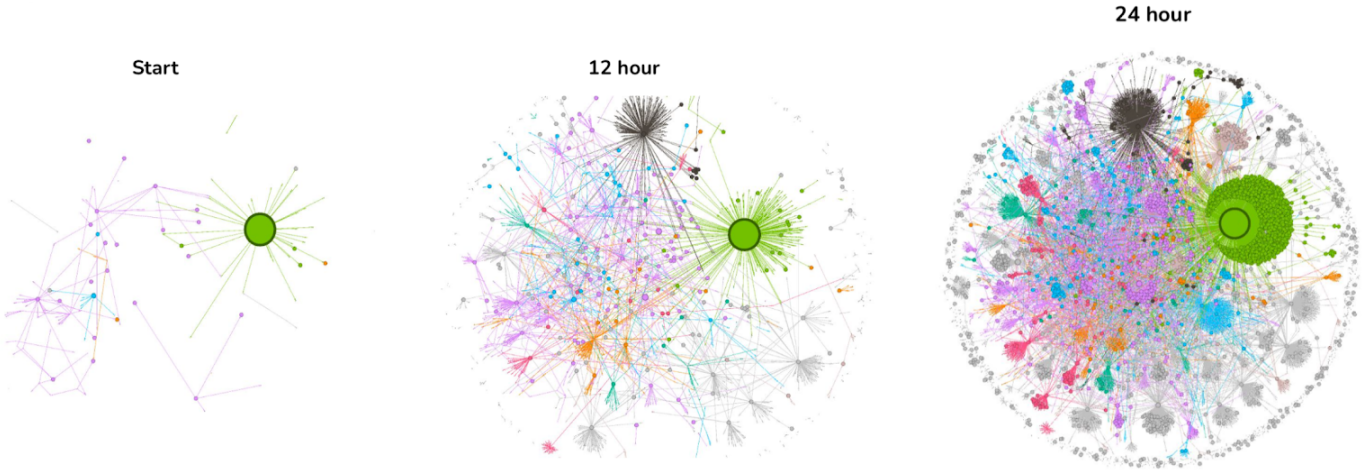



Followers network shows indeed **stronger ties** than Retweet network w/ 21 communities! (<100)

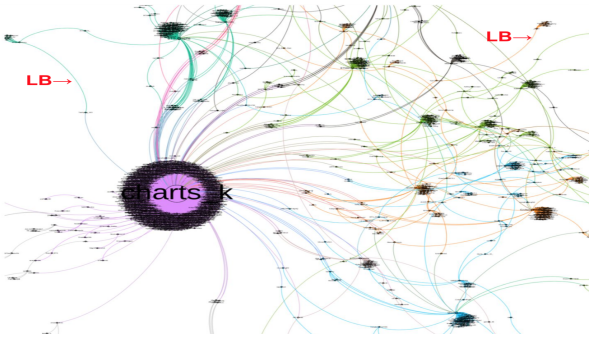
As shown left, the followers network consists of more than **five times** larger number of nodes and edges than the retweet network. However, the resulting number of connected components are 4-5, i.e. **50 times less** than the retweet network (it was 258).
This indicates that the followers network shows stronger ties than the retweet network (weak ties and many local bridges).

## III-3) Dynamic Retweet Network (Evolution over 24 Hours)

We monitor community evolution by constructing a dynamic temporal network shown below.




At the beginning, users like charts_k and BTS_History613 started spreading the message of the fundraiser campaign on Twitter. After 12 hours elapsed, the fundraiser message spread quickly with different users and different communities slowly forming.




Finally, after 24 hours elapsed, different communities like BLM and BTS Army came together to make the fundraiser a success in 24 hours. While monitoring the evolution of community formation, we noted a one time snapshot at 4:43PM-5:00PM shown left revealing a couple of single edges denoted in red arrow with label “LB”. They are considered to be a **“local bridge” (LB)** connecting small communities to the major community (a cluster centered around user “charts_k”, BTS official fan account.

## III-4) Reasons for quick spread:

To examine the underlying reasons for such a rapid spread of information in 24 hours, we carry out several structure-based detailed network analyses: 1) power law analysis using structural metrics to identify the

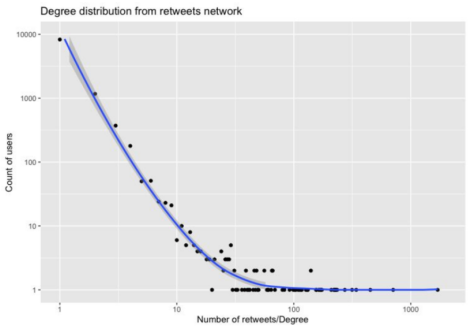
Influential nodes; 2) quantifying the user engagement metric by using the Common Neighbors Algorithm; and 3) identifying the specific nodes and its roles as local bridges connecting different communities.

### III-4a) Power Law Analysis: Influential nodes

For the retweet network, we identify the most influential user nodes based on the degree as a metric. As expected the most influential node with maximum degrees of 8,285 (accounting for 80%) is the BTS fan account. One non-fan account with 372 degrees (accounting for 0.035%) is also identified in top 5 influential nodes. The degree metric-based power law graph is shown below (left). Generally known retweet threshold is about 10, whereas in our case we obtain triple times higher threshold value of 30.

1) Retweet Network:

- Total Nodes: 10,309
- Max degree of non-fan tweet: 372
- Max degree of fan tweet: 8,285
- Most influential tweets:
  - @charts_k: 8,285
  - @OneInAnARMY: 1,168
  - @BTS_History613: 1,168
  - @GoAwayWithJae: 372

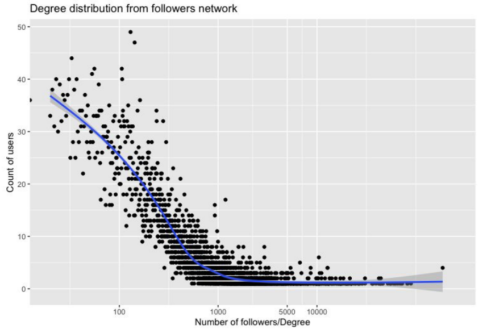



General Retweet Threshold: 10

Our Retweets Network Threshold: 30

2) Follower Network:

- Total Nodes: 10,956
- Max degree of non-fan account: 1,424,816
- Max degree of fan account: 279,062
- Most influential users:
  - @TheCut: 1,424,816
  - @charts_k: 279,062
  - @BTS_ARMYLeague: 262,027




General Threshold: 1000

Our Followers Network Threshold: 3000

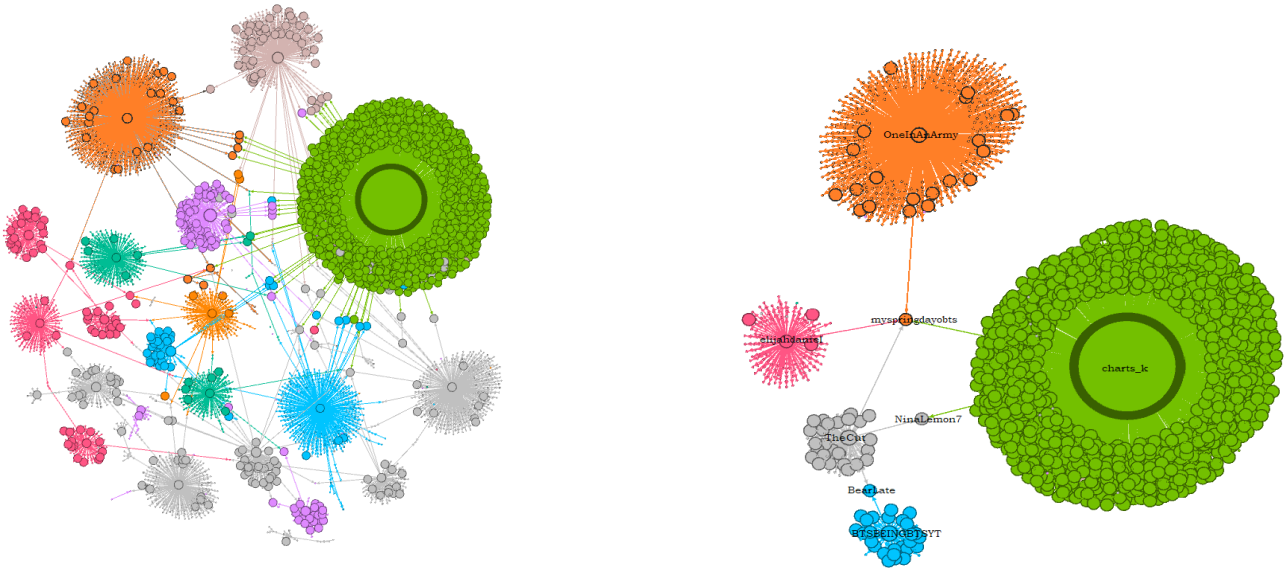
### III-4b) User Engagement Metric: Common Neighbors Algorithm

| user_name | engagement |
|---|---|
| NakitaGroenewa1 | 0.25 |
| BTS_History613 | 0.0964 |
| smilingbangtans | 0.071428571 |
| bones2blossoms | 0.066667 |
| leaoutsold | 0.056338 |

Table above shows top users with high engagement. This is fairly accurate as user "BTS_History613" is one of the sources that initially started spreading the fundraiser message and is present in the table.

### III-4c) Local Bridges Among Communities: High Connectedness

Isolated view of some of the big communities with the local bridges connecting them in the retweet network is shown below (left).




### III-4d) Bridge Nodes : Bilingual Translators

As shown above (right) the big communities around users "OneInAnArmy" (official BTS account) and "charts_k" (official BTS Fan account) are connected to the community around user "TheCut" i.e. a BLM community via local bridge node users "myspringdayobts" and "NinaLemon7". On observing some of these bridge node users it is noticed that they are bilingual translators to help improve engagement between different communities.

### III-5) Linguistic Analysis:

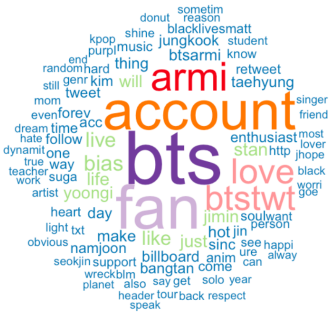
Together with the network structure-based analysis, we conduct various linguistics analysis to utilize the available text data (users bio descriptions as well as tweeter messages) in our twitter dataset: 1) frequency-based Word Clouds visualization; 2) Sentiment Analysis of tweets using Naive Bayes; 3) Word2vec embeddings and clustering; 4) more sophisticated cosine similarity-based theme clustering by Semantic Network Analysis; and 5) word-occurrence-based classification.

### III-5a) Understanding the dataset using word clouds

| User Bio Description | Tweets |
|---|---|
| Constant for a long time | Real-time |
| More likely to describe interests, likes, dislikes | More likely to describe spontaneous feelings and thoughts |
| Reflects the user | Reflects the state of the network |
| Can be used to check whether a user belong to a particular community | Can be used to understand the news propagation and overall health of a network |

Most frequent words in user_descriptions:

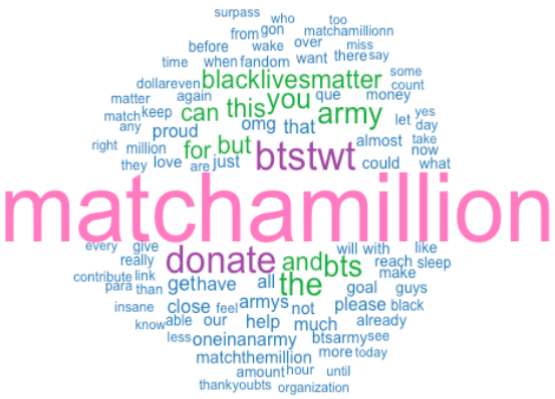
- **bts: 205**
- **fan: 164**
- **account: 144**
- armi: 97
- btstwt: 81
- love: 81
- bias 39
- live 36
- like 33
- stan 28




Most frequent words in tweets:

- **matchamillion 513**
- **donat 224**
- **btstwt 205**
- **armi 185**
- blacklivesmatt 96
- can 89
- bts 88
- proud 58
- oneinanarmi 55
- million 49

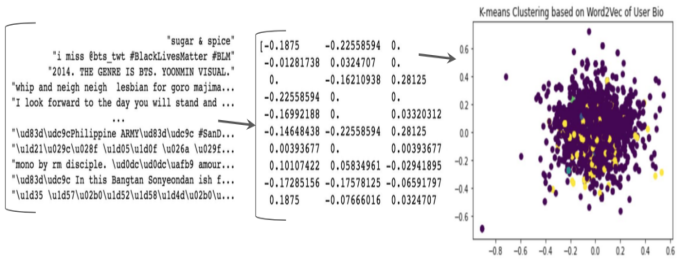

## III-5b) Sentiment Analysis of Tweets using Naive Bayes

| Positive Tweets | Negative Tweets |
|---|---|
| 537/842 ~ 63.7% | 305/842 ~ 26.3% |
| Example: oneinanarmy i have a lot of money atmso this be all i could do but happy to help matchamillion | Example: listen ... i 'm broke blacklivesmatter matchamillion |
|  |  |



## III-5c) NLP-based Behavioral/Functional Analysis

- Word Embeddings of All Tweet User's Bio Text by **Word2Vec**
- K-means Clustering with Features of Word2Vec

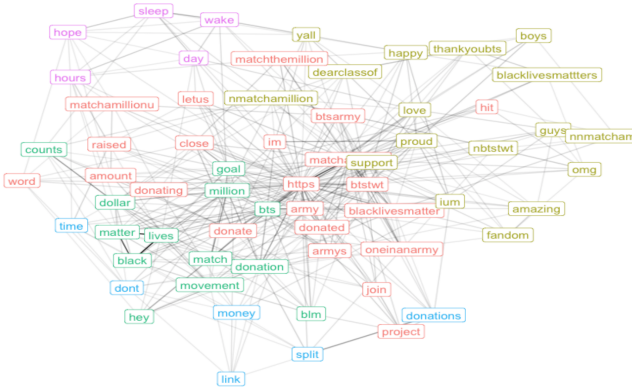



We would like to compare the structure-based clustering with a behavior-based clustering by using the twitter users information (user description text). To do so, we firstly convert the text data into numerical vectors to feed them to k-means clustering. Major difference between community detection and clustering is that in community detection, individuals are connected to others via a network of links,whereas in clustering, data points are not embedded in a network. `[mining]`
As in figure, Word2vec-based k-means clustering result shows overlapping among clusters. One reason is that the communities, detected by structure-based clustering, share similar words in their descriptions as mostly the BTS fans, thus those commonness are also reflected in the converted numeric vectors, resulting in fuzzy

clustering. Another reason is that due to twitter-specific characteristics such as including unicodes and different languages obfuscates Word2Vec algorithms.

### III-5d) Semantic Network Analysis for Themes

- Clustering on word similarity graph for tweets(1-gram):



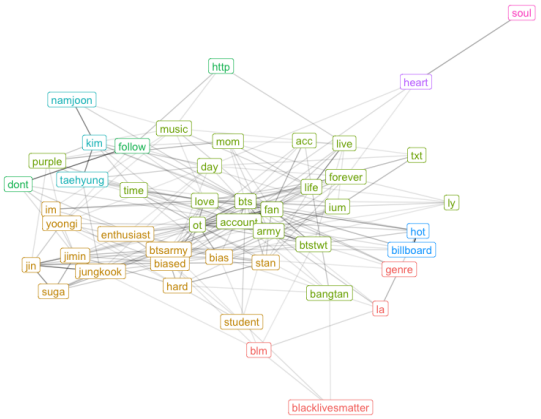

| Group 1: **A call for action** | Group 2: **Overwhelming positive support for the cause** | Group 3: **Includes the tweets relating to goals in monetary collection** | Group 4: **Includes time management in the collection** | Group 5: **Is about experienc people involved** |
|---|---|---|---|---|
| amount, army, armys, blacklivesmatter, btsarmy, btstwt, close, donate, donated, donating, hit, https, im, join, letus, word matchamillion, matchthemillion, oneinanarmy, project, raised | amazing, blacklivesmattters, boys, dearclassof, fandom, guys, happy, ium, love, bts, nmatchamillion, omg, proud, support, thankyoubts, yall | black, blm, bts, counts, dollar, donation, goal, hey, lives, match, matter, million, movement | donations, dont, link, money, split, time | day, hope, hours, slee wake |

- Clustering on word similarity graph for tweets(2-gram)


| Group 1: **Encouraging tweets for armys and links shared** | Group 2: **Personal accounts of donations** | Group 3: **Day is almost at an end, fans asking each other to pull through** |
|---|---|---|
| almost there, at k, blacklivesmatter btstwt, blacklivesmatter https, btstwt oneinanarmy, get it, i donated, in the, less than, matchamillion blacklivesmatter, matchamillion blacklivesmattters, matchamillion https, oneinanarmy btstwt, over k, than k, thank you, the matchamillion, this is, matchamillion, want to, will be | an army, btstwt matchamillion, but i, for matchamillion, going to, i could, i just, just donated, the word, to help, to matchamillion, up to, wake up, what i | a million, are so, army matchamillion, close to, matchamillion matchthemillion, matchthemillion https, so close, to the, we are, were so |

| Group 4: **Spreading news about the hashtag** | Group 5: **Pride for the community's work** | Group 6: **Encouraging others to donate politely** | Group 7: **Tagging #MatchAMillion and BTS for engagements** | Group 8: **Sending love to each other** |
|---|---|---|---|---|
| black lives, bts army, donation to, lives matter, to match, trying to, we can | i am, im so, of this, part of, proud of, proud to, so proud, to be | able to, if you, to donate, you can | btstwt https, matchamillion btstwt | i love, love you, so much |

- Clustering on word similarity graph for user bios(1-gram)




| Group 1: **Support for BLM** | Group 2: **Common army lingo** | Group 3: **Common army-adjacent lingo** | Group 4: **Directs user to a different account** | Group 5: **BTS members** | Group 6: **Relating to song charts** | Group 7: **Talking about interests** |
|---|---|---|---|---|---|---|
| **blacklivesmatter**, **blm**, genre, la | bias, biased, **btsarmy**, enthusiast, hard, im, jimin, jin, jungkook, stan, student, suga, yoongi | **acc**, **account**, army, bangtan, bts, btstwt, day, fan, forever, ium, life, live, love, ly, mom, music, ot, purple, time, txt | dont, follow, http | kim, namjoon, taehyung | billboard, hot, song | heart, soul |

### III-5e) Fan Classification: BTS Army or not?

- Dataset used: user_description
- Classification algorithm:
  - Keywords mined from various BTS Army resources and previous semantic analysis
  - Keywords assigned weights according to NER rules and ground truth
  - User_description tokens compared against keywords and assigned weights accordingly
  - Total weight for a user > 0.2 implies BTS Army account
- High number of true positives
- High number of false positives

## V. CONCLUSIONS & FUTURE WORKS

We found that the fandom has a lot of interesting features that can be further analyzed. For example, the power law thresholds are higher than for normal twitter communities. Also, we found a lot of influential nodes, whose tweets could be analyzed separately to understand how the “trend-setters” of the fandom work. Some of the analysis that we would like to try are as follows:

- Homophily of nodes using similarity metrics
- Fan classification using supervised learning
- Exploring Bridge Nodes using text analysis
- Affiliation network analysis for follower and retweet networks

## REFERENCES

[bts1] BTS ARMY Matched The Group’s $1 Million Black Lives Matter Donation, Proving The Positive Power Of Fandoms

[bts2] How the South Korean band’s fanbase – known as ARMY – raised over $1 million for the Black Lives Matter movement, mostly in just one day.

[tfhub] TensorFlow Hub

[embeddings] Embedding-as-Service : One-Stop Solution to encode sentence to fixed length vectors from various embedding techniques

[louvain] V. D. Blondel et al., Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, v2008, 10, 2008

[mining] Zafarani, R., Abbasi, M., & Liu, H. (2014). Social Media Mining: An Introduction. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139088510

[nltk] Natural Language Toolkit

[wcloud] R package wordcloud: Word Clouds

[igraph] Csardi G, Nepusz T (2006). “The igraph software package for complex network research.” InterJournal, Complex Systems, 1695. https://igraph.org.

[tm] I. Feinerer. An introduction to text mining in R. R News, 8(2):19–22, Oct. 2008. URL http://CRAN.R-project. org/doc/Rnews/